UP NEXT
ANDY LAKE
What is perfSONAR?

• An open source software collaboration led by ESnet, GEANT, Indiana University, Internet2, RNP and the University of Michigan.

• Goal is to provide network measurements between organizations to help identify and troubleshoot network issues. Most commonly these include (but are not limited to):
  – Throughput
  – Packet Loss
  – One-way latency
  – Traceroute
perfSONAR 5.0 released April 17th, 2023.

Over 50% of perfSONAR deployments currently running 5.0

Enables greater visualization and analysis capabilities through the replacement of the backend measurement storage database with OpenSearch

New pScheduler test plugins to support WiFi BSSID, 802.1X authentication, DHCP response time and more

Ubuntu 20 support added with additional OSes like
- EL8 and EL9 added in early summer
- Ubuntu 22 and Debian 11 coming soon

Looking ahead, 5.1 will focus on improving UI and leverage the changes put in place by 5.0 to add new capabilities

Image: Example of Grafana dashboard users can setup in 5.0 using our guide
Building on the Foundation

**UI:** Toolkit UI and MaDDash Beyond 5.0

**UI:** On-Demand Testing

**Tool Enhancements:** Multi-threaded iperf3

**Deployment Models:** perfSONAR on Internet2 Backbone
perfSONAR is developed by a partnership of

Toolkit UI and MaDDash Beyond 5.0

Andy Lake – ESnet – andy@es.net

perfSONAR is developed by a partnership of
Today’s Toolkit UI
Today’s MaDDash UI
Thinking about a new interface

- User Requirements
  - Users need to still have ability to view test results of local tests
  - Authentication required to see additional system stats, configure tests, and configure LS registration
    - Whole subset of requirements around configuring test that won’t cover in this deck
  - System information needs to be available via an API (see current JSON service)

- Dev Requirements
  - Devs need to get rid of old perl cgi scripts that are increasingly difficult to maintain
  - Ditto old Javascript
  - Consolidate redundant interfaces (e.g. PWA and Toolkit Test Config)
  - Stop discovering system stats that differ across OSes, when other things have this solved (e.g. node_exporter)

- Other nice to haves
  - Users have greater flexibility in building own views of data
  - Tighter integration with MaDDash and main toolkit UI
  - We don’t have a lot of UI devs, if there is a way to make easier to maintain without being JS expert, that is good thing.
What is Grafana?

• Grafana is an **open-source platform** for exploring data from a variety of sources
• It has a few key features
  • It’s **multi-data source**
  • It has a bunch of **built-in visualizations** that **don’t require you to be a Javascript developer** to use
  • It has a **plugin framework** for all of the above so they can be extended and a process for becoming official plugins
• More Info: [https://grafana.com/docs/grafana/latest/introduction/](https://grafana.com/docs/grafana/latest/introduction/)
Custom Plugins by NetSage and ESnet
New UI: Focus on Measurements
New UI: Enhancing the Fundamentals
New UI: Instrumentation
New UI: Data Correlation

iperf3 overlay on host metrics
New UI: MaDDash Integration
New UI: Customization
Check out our booth for a live demo!
UP NEXT
SARAH LARSEN
Multi-Threaded iperf3

First There was One, Now There are Many

Bruce Mah and Sarah Larsen, ESnet
What is iperf3?

- iperf3 is an open-source tool that measures network traffic performance between a client and server
- It was designed to be used in perfSONAR, but can also be used on its own
- Linux, MacOS, and FreeBSD are all officially supported
Current State

• Before multi-threading, iperf3 was capable of 30-50 Gbps, with single stream TCP, possibly more with tuning

• Many links in ESnet6 are faster than iperf3
  ○ Site connections: N x 100G
  ○ Backbone: 400G +

• So what if we want to support connections with higher bandwidth? How do we get more throughput?

• The problem: Adding more parallel connections doesn’t increase throughput
What are we doing about it?

- Adding multi-threading to iperf3
  - Goal: Use multiple CPU cores
  - Example command: `iperf3 --parallel 3`
- Note: currently this refers to multiple parallel streams, but in the new multi-threaded iperf, this will represent the number of threads. This is because in the new version each stream will get its own thread.
Performance Testing

● Test Setup:
  ○ Computers:
    ■ Back-to-back, no routers
    ■ 200 Gbps link
    ■ 16 cores, 2 packages, 32 cores total
    ■ More cores than parallel streams
    ■ Mellanox ConnectX6
    ■ ~3 GHz AMD CPU
Effects on Performance

iperf3-mt has significant improvements in throughput performance over iperf3.
How does it compare to iperf2?

Multi-threaded iperf3 has similar throughput results to iperf2

![Graph showing throughput comparison between iperf3 and iperf2](image-url)
What’s Next?

- Testing on ESnet perfSONAR nodes and integration with pscheduler
- Further analysis with how threading impacts performance, virtual machines vs hardware
- Ensure correctness of threading prototype
- Experiments with tuning
When?

“Soon”
But seriously Q4 2023/Q1 2024

We’re going to test it on ESnet perfSONAR hosts before releasing the final version.

Any interest in testing or want the current working version:
https://github.com/esnet/iperf/tree/mt
Summary

The new multi-threaded iperf3 can better test performance of faster paths.

GitHub: https://github.com/esnet/iperf
Multi-threaded: https://github.com/esnet/iperf/tree/mt
Contact email: iperf@es.net
Questions?

I HAZ A QUESTION
perfSONAR on the Internet2 Backbone

Mark Feit – Internet2 / perfSONAR Development Team
mfeit@internet2.edu
WE DID A THING
A Plethora of Public perfSONAR Points in PoPs
Using the Internet2 Public perfSONAR Nodes

**PoP.ps.internet2.edu**

- IPv4 and IPv6 available

- Log into a perfSONAR system (yours, not ours) and run a task with pScheduler

```
pscheduler task throughput --dest PoP.ps.internet2.edu
```
<table>
<thead>
<tr>
<th>City</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Albany</td>
<td>alba</td>
</tr>
<tr>
<td>Ashburn</td>
<td>ashb</td>
</tr>
<tr>
<td>Atlanta</td>
<td>atla</td>
</tr>
<tr>
<td>Boise*</td>
<td>bois</td>
</tr>
<tr>
<td>Boston**</td>
<td>bost</td>
</tr>
<tr>
<td>Charlotte</td>
<td>char</td>
</tr>
<tr>
<td>Chicago</td>
<td>chic</td>
</tr>
<tr>
<td>Chicago</td>
<td>eqch</td>
</tr>
<tr>
<td>Cincinnati</td>
<td>cinc</td>
</tr>
<tr>
<td>Cleveland</td>
<td>clev</td>
</tr>
<tr>
<td>Dallas</td>
<td>dall</td>
</tr>
<tr>
<td>Dallas</td>
<td>dall3</td>
</tr>
<tr>
<td>Denver</td>
<td>denv</td>
</tr>
<tr>
<td>El Paso</td>
<td>elpa</td>
</tr>
<tr>
<td>Fargo**</td>
<td>farg</td>
</tr>
<tr>
<td>Hartford</td>
<td>harrt2</td>
</tr>
<tr>
<td>Houston</td>
<td>houh</td>
</tr>
<tr>
<td>Houston</td>
<td>hous</td>
</tr>
<tr>
<td>Indianapolis</td>
<td>indi</td>
</tr>
<tr>
<td>Jackson</td>
<td>jcsn</td>
</tr>
<tr>
<td>Jacksonville</td>
<td>jack</td>
</tr>
<tr>
<td>Kansas City</td>
<td>kans</td>
</tr>
<tr>
<td>Las Vegas</td>
<td>lasv</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>losa</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>losa2</td>
</tr>
<tr>
<td>Louisville</td>
<td>loui</td>
</tr>
<tr>
<td>Minneapolis</td>
<td>minn</td>
</tr>
<tr>
<td>Missoula</td>
<td>miss2</td>
</tr>
<tr>
<td>Nashville</td>
<td>nash</td>
</tr>
<tr>
<td>New York</td>
<td>newy2</td>
</tr>
<tr>
<td>New York</td>
<td>newy32aoa</td>
</tr>
<tr>
<td>Pensacola*</td>
<td>pens</td>
</tr>
<tr>
<td>Philadelphia</td>
<td>phil</td>
</tr>
<tr>
<td>Phoenix</td>
<td>phoe</td>
</tr>
<tr>
<td>Pittsburgh</td>
<td>pitt</td>
</tr>
<tr>
<td>Portland</td>
<td>port</td>
</tr>
<tr>
<td>Raleigh</td>
<td>rale</td>
</tr>
<tr>
<td>Reno</td>
<td>reno</td>
</tr>
<tr>
<td>Sacramento</td>
<td>sacr</td>
</tr>
<tr>
<td>Salt Lake City</td>
<td>salt</td>
</tr>
<tr>
<td>San Jose</td>
<td>sanj</td>
</tr>
<tr>
<td>Seattle*</td>
<td>seat</td>
</tr>
<tr>
<td>Sunnyvale</td>
<td>sunn</td>
</tr>
<tr>
<td>Toledo</td>
<td>tole2</td>
</tr>
<tr>
<td>Tucson</td>
<td>tucs</td>
</tr>
<tr>
<td>Tulsa</td>
<td>tuls</td>
</tr>
<tr>
<td>Washington</td>
<td>wash</td>
</tr>
</tbody>
</table>

*Not yet in service  **Future
## Topology

<table>
<thead>
<tr>
<th>PoP Type</th>
<th>Router Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed</td>
<td></td>
</tr>
<tr>
<td>Multi-Degree</td>
<td>Core*</td>
</tr>
<tr>
<td>Interconnect</td>
<td></td>
</tr>
<tr>
<td>Split Interconnect</td>
<td>Aggregation*</td>
</tr>
</tbody>
</table>

*First where more than one is installed
Network Reachability

Now
R&E + I2PX

Later
Elsewhere
Initial Administrative Limitations

• Caps on **throughput** test bandwidth
  • R&E* 10 Gb/s
  • Elsewhere 1 Gb/s
  • Higher bandwidth considered on a case-by-case basis

• No, **disk-to-disk, s3throughput, idleex or wifibssid** tests

• Tests with a **duration** parameter are limited to one minute

• Repeating tests
  – Not more-frequently than once per hour
  – Must wrap up within 24 hours
  – repeat-until parameter not allowed

• Testing priorities
  – Internet2 Internal
  – R&E Networks
  – Everyone Else

• These limits will be refined periodically to make sure the community’s needs are being met.

*Determined using ESnet’s R&E network list: [http://stats.es.net/sample_configs/pscheduler/ren](http://stats.es.net/sample_configs/pscheduler/ren)
Beta Period

- Now through January
- Feature set close to production
- Feedback is appreciated
  pas@internet2.edu
Beta Will Be Beta

• Working through some teething with the NICs in some PoPs
  • Systems disappear from the network

• Ongoing experimentation on systems in CLEV, PHOE and TUSC PoPs
Performance Tuning

New OS
+ New Kernel
+ New NIC Driver
= New System Tunings

• Higher-speed throughput requires additional test parameters to run well
THE SYSTEMS
A Chicken System in Every Pot PoP

- Dell R6515 (NGI Buildout) and R6615 (Later)
- AMD EPYC 7402P 24 Cores / 48 Threads at 2.8 GHz
- 128 GiB RAM

- Broadcom 2x 10 GbE
- Mellanox Connect-X 5 2x 100 GbE

- AlmaLinux 9

Connections to routers vary by PoP type.
Host Architecture

- OOB Management Interface
- Lights-Out Management
- OS Management Interface
- Host OS
- Container Environment (Podman or Docker)
- perfSONAR Container
- Other Projects
- Measurement Interface
**Special Secret Sauce: The `macvlan` Network Driver**

- Binds a host interface directly into a container
- Bypasses additional container networking code
- Negligible performance difference vs. bare metal
- No address assigned on the host
  - Prevents external access to the OS
Resources for Performance

- **100 GbE Interface Containers**
  - 12 Dedicated CPU Cores (Threads)
  - 32 GiB Dedicated RAM
  - IRQ Affinity (Tuning)
  - 93+ Gb/s

- **10 GbE Interface Containers**
  - Shared CPU Cores
  - Shared Memory
  - No special performance tuning
Deployment Technology Stack(s)

- Single data set with detailed information on PoPs, systems, interfaces and networks
  - System kickstart files
  - Ansible
  - Salt
  - Assorted shell scripts
  - DNS records
  - Internal proxy configuration and ACLs

- Ultimate goal is to use Salt
Why so many different technologies?

- The tools are buggy.
- IPv4 /31s used for point-to-point connections
  - Halves address use compared to /30s
  - See RFC 3021
- Docker couldn’t handle those at all
  - Patch submitted in 2021, released in Docker 23.0.0 (2022).
- Podman is fine with them but its web API isn’t.
The Long and Winding Road

• Began with Ansible for expediency
  • Problems with networks being re-created with each run
  • Destroyed/rebuilt the container. *No bueno.*

• Tried Salt
  • Stymied by the /31 problem

• Ended up with a set of shell scripts
  • Tied together for single-command provisioning of the entire system
  • Small bites that can be easily converted to Ansible or Salt
Future Development: Deployment Kit

• Will be derived from deployments at Internet2 and elsewhere
  • Based around Ansible
  • Minimal manual perfSONAR host configuration
  • Driven by data: Configure and go
  • Container-per-interface model (Plain or VLANs)

• Initial version targets Debian
  • EL to follow
Thanks!

Public perfSONAR

**PoP.ps.internet2.edu**

**Feedback**

pas@internet2.edu
UP NEXT
LÆTITIA DELUAUX
perfSONAR in GÉANT

2023 edition

Lætitia A Delvaux, PSNC / GÉANT Project
GN5-1 WP6T3 Task leader

TechEX23, Minneapolis, MN, USA
20 September 2023

Public (PU)
Multiple perfSONAR activities in GÉANT

- Lookup Service dashboards
- perfSONAR deployments
- Microdep integration
- On-demand perfSONAR Graphical User Interface (psGUI)
Lookup Service dashboards

- Display, filter and search the content of the Lookup Service
- Grafana 8 based
  - Filtering on text, domains, communities
  - Stats on hosts and services, maps
- [https://stats.perfsonar.net](https://stats.perfsonar.net)
  - Replaces ESnet Service Directory

Next steps:
- Port to Grafana 9
- Filter on multiple values
perfSONAR deployments in the GÉANT network (1/2)

- 10 public deployments on the core network: [https://network.geant.org/perfsonar/](https://network.geant.org/perfsonar/)
perfSONAR deployments in the GÉANT network (2/2)

• Performance Measurement Platform (PMP)
  • Small nodes (Intel NUC) and VM
  • Deployed in GÉANT partners organisations

• Measurements
  • Diverse set of measurements
  • Regularly to GÉANT core network
  • Verify GÉANT access links
  • International connections (ESnet, Internet2, RNP, ...)
  • 2nd tiers: University networks

• https://pmp-central.geant.org
Microdep integration with perfSONAR (1/2)

- Microdep is a packet loss analysis and visualisation tool
  - Spotting packet gaps, micro failures, ~10 packets loss
  - Using 100 packet/sec probes
  - Traceroutes and ICMP response monitoring

- Realtime event analysis:
  - Packet-loss (gaps)
  - Queues (jitter)
  - Route failures and changes (traceroute)
  - Joint event anomaly and alarms (ELK and ML)
Microdep integration with perfSONAR (2/2)

- Using perfSONAR to generate probes
  - OWAMP for paced packets
  - Traceroute
  - Rely on 2000+ public perfSONAR hosts
  - Use pSConfig and pScheduler

- Adding a data pipeline to
  - Analyse packet gaps
  - Store history for further analysis

- Next steps:
  - Package and bundle with pS
On-demand perfSONAR Graphical User Interface (psGUI) (1/2)

- GUI to drive perfSONAR / pScheduler
- Use case:
  - MaDDash setup, grids, regular measurements
  - Want to do a one off, on-demand additional test
  - List of pS nodes coming from pSConfig file, MaDDash grids
- GUI list all possible tests and main parameters:
On-demand perfSONAR Graphical User Interface (psGUI) (2/2)

• Results:

• Packaged as a Docker Image to be built

• [https://github.com/perfsonar/psgui](https://github.com/perfsonar/psgui)
Thank You

Contact: perfsonar@lists.geant.org
THANKS FOR JOINING US!