perfSONAR

pShooter

A tool for automating troubleshooting with perfSONAR

Mark Feit mfeit@internet2.edu

European perfSONAR Workshop June 5-6, 2019

ESnet ENERGY SCIENCES NETWORK GÉANT INDIANA UNIVERSITY INTERNET2 UNIVERSITY OF MICHIGAN

pShooter

- Automated, **p**erfSONAR-based performance trouble**shooter**

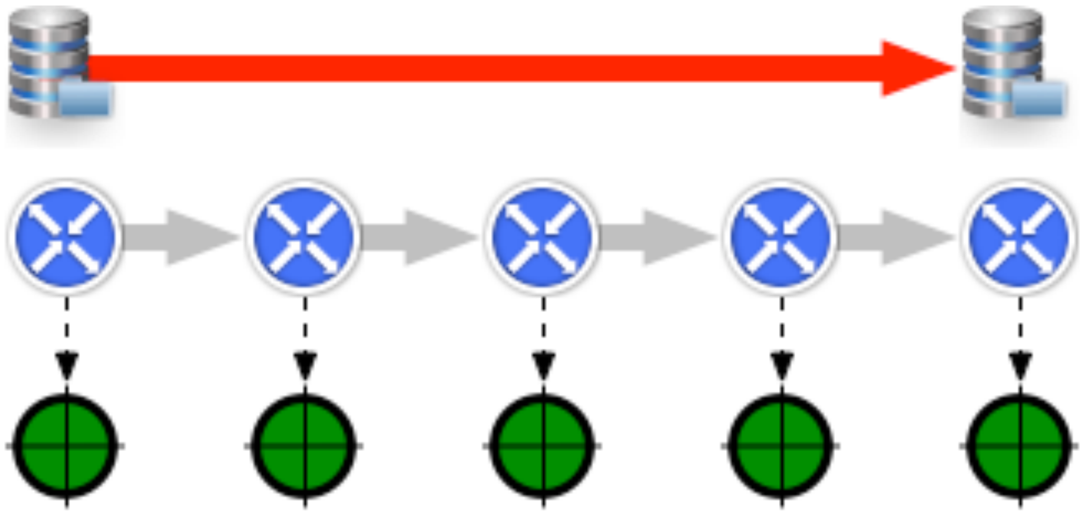
- Came about as a result of some discussions at Internet2 in early 2018.

Two Topics

- Adding performance data to existing visual traceroute tools
- Identifying perfSONAR nodes near points along a path
  - Initially for locating stored measurements

Locating perfSONAR Nodes

The Troubleshooting Process, Part 1



Observe poor performance

Trace the path between hosts

Identify nearby perfSONAR nodes

perfSONAR

Where to Make Measurements?

- Lookup Service is geographic, not topological.

- Candidate nodes and the paths between them and the point along the path have to be checked out.
    - Recursive problem: Need to find other perfSONAR nodes to test these paths.

ESnet ENERGY SCIENCES NETWORK GÉANT INDIANA UNIVERSITY INTERNET2 UNIVERSITY OF MICHIGAN

The Question We Want Answered

Q: My traffic passes through **`198.51.100.26`**. What perfSONAR node should I use for testing to that point along the path?

A: Use **`perf2.lnd.foo.net`**.

The Question We Want Answered

- Network operators know the answer.
- Automate the asking and answering.

A Low-Effort Solution

DNS

Yeah. We went there.

DNS to Distribute Nearest-Node Information

- **Why use DNS?**

• Ubiquitous	*Everybody runs a DNS server*
• Reliable	*Robust infrastructure deployed worldwide*
• Cached	*Helps mitigate outages of authoritative servers*
• Available	*Rarely blocked, almost always at least proxied*

- **Better than a single, static directory**
  - No central resources required
  - Network operators control their own DNS servers
  - Ability to give different answers to different questioners (e.g., BIND views)

How's it Done?

• Start with a path IP	`198.51.100.26`
• Reverse resolve to FQDN	`e6-2.lnd.foo.net`
• Add prefix	`_ipv4._perfsonar.e6-2.lnd.foo.net`

• Resolve to **`TXT`** record	`{ "pscheduler":`
	`"perf2.lnd.foo.net" }`

- No record found at any stage means no operator-recommended perfSONAR node.

Special Provisions for Special Situations

- **Single- and dual-stack IP environments**
  - Prefix for IPv4 **`_ipv4._perfsonar`**
  - Prefix for IPv6 **`_ipv6._perfsonar`**
  - Last-resort, non-specific prefix **`_perfsonar`**
  - Prefix avoids colliding with site’s other **`TXT`** records

- **Redirection of queries to a URL**
  - 255-byte length limit of **`TXT`** records on some DNS servers
  - Allows for dynamically-generated answers

How’s it Done?

- pShooter will look for records in shorter versions of the hostname so it doesn’t have to be done per-interface:

`_ipv4._perfsonar.e6-2.lnd.foo.net`	Interface
`_ipv4._perfsonar.lnd.foo.net`	PoP
`_ipv4._perfsonar.foo.net`	Entire provider
`_ipv4._perfsonar.net`	gTLD

perfSONAR

The Complete Skinny

- The entire scheme is documented.
- URL at the end of the presentation.

perfSONAR

Be an Early Adopter

- Fewer calls to your NOC…
  - …asking about your perfSONAR nodes
  - …wrongly pointing fingers for network problems
- Less perfSONAR time consumed on fishing expeditions
- Often easy to generate the required data
- We may provide scripts for generating BIND zones from CSV or JSON.

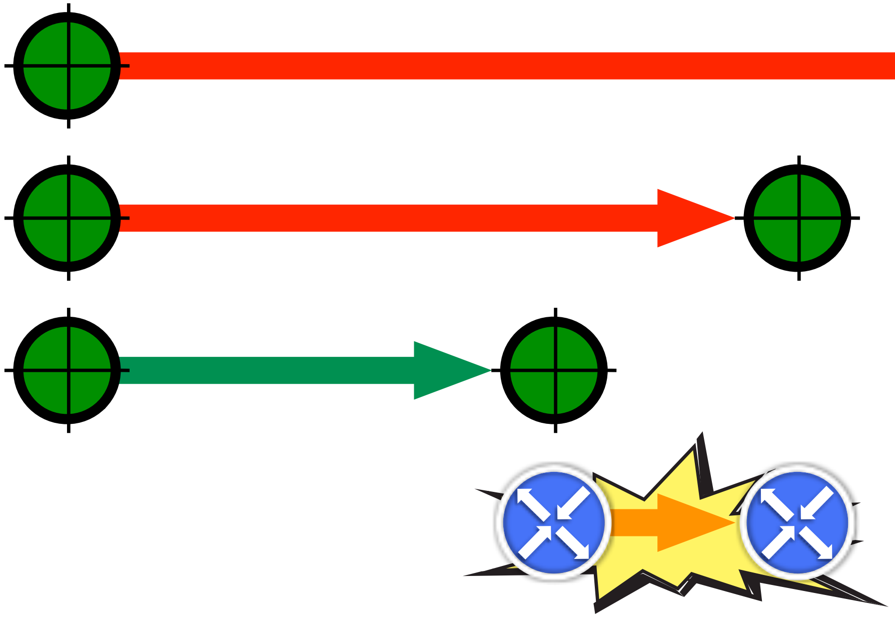
ESnet ENERGY SCIENCES NETWORK GÉANT INDIANA UNIVERSITY INTERNET2 UNIVERSITY OF MICHIGAN

perfSONAR

Automatic Troubleshooting

ESnet ENERGY SCIENCES NETWORK GÉANT INDIANA UNIVERSITY INTERNET2 UNIVERSITY OF MICHIGAN

The Troubleshooting Process, Part 2



Test entire path

Test shorter path

Repeat until problem disappears

Investigate likely sources

Going About It

- We know the path in question.
- We know the measurement of interest.
- We have a way to find perfSONAR nodes along the path.

- Measuring along the path can be automated.

- Enter ***pShooter***.

pShooter

- Web service: request goes in, results come out (eventually)
- Uses the pScheduler REST API to get the work done
- Lots of potential for integration with other systems

- Doesn’t make for a visually-interesting demo

Input JSON: Top Level

```
{
   "path":  …,           Path to be tested
   "test":  …,           Measurement to be performed to each hop
   "dns":  …,            DNS augmentation
   "callback":  …        URL to be fetched when complete
}
```

`POST` to **`https://hostname/pshooter/tasks`**.

Returns a URL for the task. (pShooter task, not pScheduler task)

Input JSON: Path

```
"path":  "198.51.100.26"
```

pShooter traces the path

```
"path": [ "192.0.2.17",
          "203.0.113.149",
          "198.51.100.26" ]
```

Supply your own path

- Addresses bound to local interfaces are removed
- pShooter tests from itself each of the addresses that remains

Input JSON: Test

```
"test": {
  "type": "rtt",
  "spec": {
    "source": "__A__",
    "dest": "__Z__",
    "count": 5
  }
}
```

Test spec as would be exported by the pScheduler CLI

*Special marker for near end**

Special marker for far end

**Optional in most cases*

Input JSON: DNS Augmentation (Optional)

This data is consulted first, then DNS.

```
"dns": {
  "arpa": {
    "in-addr": {
      "198.51.100.26": { "PTR": "e6-2.lnd.foo.net" }
    }
  }
  "foo.net": {
    "_ipv4._perfsonar.e6-2.lnd": {
      "TXT": "{\"pscheduler\": \"perfsonar.lnd.foo.net.\"}"
    }
  }
}
```

**Optional in most cases*

Input JSON: Callback (Optional)

```
"callback": {
  "_href": "https://netops.foo.net/pshooter-complete",
    "_params": {
      "id": "281apple",
      "authenticator": "ou812”
    },
  "_headers": {
    "X-Highway-To": "danger-zone",
  },
  "retry-policy": [
    { "attempts": 2, "wait": "PT1S" },
    { "attempts": 2, "wait": "PT2S" },
    { "attempts": 2, "wait": "PT3S" }
  ]
}
```

Same as for pScheduler archive specifications

What pShooter Does On the Inside

- Finds the nearest perfSONAR node for each point along the path
  - DNS method or direct check
- Runs tests from itself to each point along the path where pScheduler is available
- Single-participant tests (e.g., `rtt`, `latency`) are run with the hop IP as the destination if pScheduler is not available.
- Collects the results and diagnostic information
- Produces a final result at the task's endpoint

What pShooter Produces

- Blob of JSON.
- Results of each test performed
  - JSON for machines, plain text and HTML for humans
- Information about the test (Mostly what went in)
- Hop and perfSONAR host information
- Other diagnostic information

A Note About Limits

- pScheduler enforces limits based on the requester.
  - Requests from pShooter would appear to come from the local host.
  - The local host is usually allowed to do pretty much anything.
- 4.2.0 will have a feature allowing local applications that properly authenticate themselves to pScheduler to name the original requester and have it believed.
- pShooter will pass the requester information on to pScheduler

Release and Other Evil Plans

- pShooter depends on features in pScheduler 4.2.0
- Beta sometime after 4.2.0.
- Full production when it's had some to be tested in the field.

perfSONAR

Thanks!

- **Brief: *DNS as a Locator Service for Nearby perfSONAR Nodes***
  - https://internet2.app.box.com/v/pshooter-dns

- **More Information -** Mark Feit, mfeit@internet2.edu.

ESnet ENERGY SCIENCES NETWORK GÉANT INDIANA UNIVERSITY INTERNET2 UNIVERSITY OF MICHIGAN