

perfSONAR

Introducing *pScheduler* *perfSONAR's New Scheduler*

Presented by the perfSONAR Project

<http://www.perfsonar.net>

GÉANT eduPERT Training — November, 2016

This document is a result of work by the perfSONAR Project (<http://www.perfsonar.net>) and is licensed under CC BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0/>).



pScheduler

- *pScheduler* is new software for scheduling and supervising measurements.
- Successor to the Bandwidth Test Controller (BWCTL)

Why replace BWCTL?

- BWCTL has served perfSONAR well but is showing its age.
 - Maintenance and expansion have become difficult.
- Demand for new features and additional sophistication called for a clean-slate implementation built with an eye toward the future.

We Get Letters

- **Tests and Tools.** Can perfSONAR measure _____?
- **Schedule.** What ran? What's running? What's planned to run?
- **Status and Diagnostics.** What happened? Why do I have to log in and pick through the logs?

We Get Letters

- **Regular Testing.** Can it work with _____? Can it be better? Less kludgy?
- **Archiving.** Can results be sent to _____? Reliably?
- **Persistence.** Why does the schedule not survive a reboot?
- **Open Issues in GitHub.** Lots of them.

Our Goals

1. Look at every aspect of measurement through as many lenses as possible.
2. Design something adaptable to new needs.
3. Build it to last a long time.

perfs-SONAR

NOMENCLATURE AND CONCEPTS

pScheduler Terminology

- ***Test*** Definition of a measurement
- ***Participant*** Point involved in the test
- ***Tool*** Program that does the measuring
- ***Run*** The making of a measurement
- ***Result*** The end product of a run
- ***Archiver*** A way to send result to storage
- ***Task*** A job for pScheduler to do

Tests in the Abstract

- Measurement has been abstracted.
 - Test parameters *Standard format*
 - Result *for each test.*
- Tests describe the parameters, tools do the measuring.
- Tool-specific features are accommodated.

perfs-SONAR

WHAT'S NEW

Extensibility

- Many of the things BWCTL and other parts of perfSONAR used to do internally have been delegated to plugins:
 - Tests *Measure new things.*
 - Tools *Use new things to do measurements.*
 - Archivers *Send results where you want.*
 - Other things to come in the future

Extensibility

- Documented API – There will be training.
- Minimal involvement from core development team required
 - We ask early adopters to work with us
- Support for niche applications
 - Do more with your perfSONAR deployments
- Shorter development cycle

Tests and Tools

- **Throughput** Iperf2, Iperf3, Nuttcp
- **Latency** OWAMP, Powstream
- **Round-Trip** Ping
- **Trace** Traceroute, tracepath, Paris Traceroute
- **Development and Demo:**
 - **Simplestream** Simplestreamer
 - **Idle** Sleep, Snooze

Tool Selection

- Just One **owamp**
- Preferred Order **iperf3, iperf2**
- Automatic
 - Tool availability and hard-wired preferences (e.g., **traceroute** over **tracpath**).
 - Allows smooth deprecation of older tools and easy introduction of new ones.

Archivers

- **Shipping** Esmond, Syslog
 - **Dev and Demo** Bit Bucket, Failer
 - **Near Future** HTTP, RabbitMQ, Others
-
- Archivers can be selected per task or applied globally to every result.

Repetitive Testing

- Baked into the core of the system.
- Now part of scheduling runs for *every* task.
- All tasks work the same way:
 - Some repeat, some don't.
 - All test features are available, repeating or not.
 - No special development required for new tests or tools.

Limit System

- All new – There will be training
- Four Phases
 - Identification *Who's asking?*
 - Classification *What group(s)?*
 - Limits *What are the restrictions?*
 - Application *Force test to conform to...*

Limit System

- **Identification** *Who's asking?*
 - CIDRs (explicit list, download/update), Reverse DNS, Bogons/Martians, Hints (IP, protocol), Forced

- **Classification** *What groups?*
 - E.g., bogons, selected IPs and a list downloaded from IDS fall into a ***hostile*** classification.
 - IPs in a downloaded list of R&E CIDRs fall into a ***research*** classification.

Limit System

- **Limits** *What are the restrictions?*
 - Test type and parameters, schedule attributes
- **Applications** *How are the limits applied?*
 - Deny requests in the *hostile* classification
 - Allow requests in the *research* classification that pass limits *research-throughput* or *inert-tests*

Computer and Human Interfaces

- REST
 - All other interfaces operate through this one.
 - 100% JavaScript Object Notation (JSON)
- Command-Line
 - Not identical to **bw*** family of commands but not a difficult adjustment
- GUI (Future)

CLI Features

- Export and import tasks to/from JSON
 - Edit parameters on the fly
- Watch repeating tasks as they run
- Retrieve previously-stored results
- Query the schedule

Other Features

- Per-task reference data
 - Add arbitrary JSON to any task.
 - Query against this and other task parameters via the REST API
- Extensive diagnostics and debug
 - Reduced email volleys required to solve problems

Backward Compatibility

- pScheduler is incompatible with BWCTL
- BWCTL will be kept at least through fall, 2017
- During the transition:
 - Tool plugins that can talk to BWCTL-only systems for throughput, latency, round-trip time and traceroute
 - perfSONAR 3.x Regular Testing and MeshConfig have been adapted to work with pScheduler.

perfs-SONAR

TECHNICAL FEATURES

Relational Database Backend

- PostgreSQL 9.5 – Selected for scale up/down, maturity and full feature set
- Heavy lifting through queries instead of hand-rolled code
- Integrity features cut down on bugs and consistency problems induced by software bugs and system crashes.
- Upgrade process is 100% transactional.

Plug-in Architecture

- API for extending parts of the system
- No modifications to pScheduler needed
- Process boundary
 - Develop in any language
 - All communication is JSON via standard I/O
 - pScheduler is protected from plug-in-related crashes

Supporting Code

- Written in Python 2.x
 - Well-used and well-understood by the community
- Recommended language for plug-ins
 - Not required
 - `pscheduler` Python module will make common utilities available to speed development (e.g., inhaling JSON, standard exit methods, safe running of external processes).

perfs-SONAR

A QUICK LOOK AT THE CLI

```
% pscheduler task throughput \  
    --dest ps.bar.net --duration PT15S \  
    --interval PT3S
```

Submitting task...

Task URL:

```
https://ps.foo.net/pscheduler/tasks/9d30dbf7-746f-4fef-9640-814896b2b8a5
```

Fetching first run...

Next run:

```
https://ps.foo.net/pscheduler/tasks/9d30dbf7-746f-4fef-9640-814896b2b8a5/runs/a785d67e-08af-4614-ab91-593425ee04d5
```

Starts 2016-09-12T01:29:06Z (~3 seconds)

Ends 2016-09-12T01:29:26Z (~19 seconds)

Waiting for result...

* Stream ID 4

Interval	Throughput	Retransmits	Current Window
0.0 - 3.0	854.60 Mbps	630	1.40 MBytes
3.0 - 6.0	599.98 Mbps	12	849.98 KBytes
6.0 - 9.0	551.65 Mbps	0	923.82 KBytes
9.0 - 12.0	575.41 Mbps	0	931.06 KBytes
12.0 - 15.0	464.74 Mbps	3	750.06 KBytes

Summary

Interval	Throughput	Retransmits
0.0 - 15.0	609.29 Mbps	645

perfs-SONAR

FINALLY...

Bear With Us


- pScheduler represents a very significant change.
- We expect there will be some teething.

Training, Seminars and Short Subjects

- pScheduler Overview (Long Version)
 - Command-Line Interface
 - Plugin Development
 - Archiving *More to come.*
-
- Announcements on **perfsnar-users**

Feedback

- Your comments, questions and ideas are the source of many pScheduler features.
 - Please keep them coming.
- **perfSONAR Developer Mailing List:**
<https://lists.internet2.edu/sympa/info/perfsonar-developer>
- **Email me directly:** `mfeit@internet2.edu`

perfs--NAR

The End

Questions?

Comments?