The pScheduler Command-Line Interface

ASTRON perfSONAR training

Antoine Delvaux, PSNC, antoine.delvaux@man.poznan.pl
Szymon Trocha, szymon.trocha@man.poznan.pl
24-26 September 2018

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/).

What is pScheduler?

- Software for scheduling, supervising and archiving measurements.

- Complete replacement for the Bandwidth Test Controller (BWCTL) as a component of perfSONAR

© 2018, http://www.perfsonar.net

Why replace BWCTL?

- Parts of it are becoming creaky with age.
- Architecture makes many community-requested features difficult to implement.
- After extensive evaluation, a clean slate with an eye toward the future was determined to be the best option.

© 2018, http://www.perfsonar.net

Highlighted Improvements

- Full support for all tools supported by BWCTL and more
- Visibility into prior, current and future activities
- Measurement diagnostics provided with results
- Full-featured, repeating testing for all measurement types baked into the core of the system
- More-powerful system for imposing policy-based limits on users
- Reliable archiving (with multiple archivers, including Esmond, RabbitMQ and HTTP)

© 2018, http://www.perfsonar.net

Major Improvement: Extensibility

- Plug-in system allows integration of new…
  - Tests — *Things to measure*
  - Tools — *Things to do the measurements*
  - Archivers — *Ways to dispose of results*
- Well-documented API
- Easily brings new applications into the perfSONAR fold
- Core development team doesn't need to be involved other than in an advisory role

© 2018, http://www.perfsonar.net

Test Abstraction

- pScheduler abstracts the tests you do from the tools that do the measurements.
  - **`throughput`** not **`bwctl`** or **`iperf`**
  - **`latency`** not **`owamp`**
  - **`rtt`** not **`ping`**
  - **`trace`** not **`traceroute`**
- There are provisions for tool-specific features and selection of specific tools.

© 2018, http://www.perfsonar.net

The Basics

© 2018, http://www.perfsonar.net

Front End

- pScheduler is operated using a single command-line program:

`pscheduler`

- Autocompletes easily on most systems:

`psc` *Tab*

© 2018, http://www.perfsonar.net

Command Format

- All commands follow the same format:

```
pscheduler command [ arg … ]
```


© 2018, http://www.perfsonar.net

Getting Help

- The **`--help`** switch can be used at any point along the command line for assistance:

```
pscheduler --help
pscheduler command --help
```


© 2018, http://www.perfsonar.net

Task Commands

- **`task`** – Give pScheduler a task that consists of making one or more measurements (*runs*).
- **`result`**– Fetch and display the results of a single, previously-concluded run by its URL.
- **`watch`** – Attach to a task identified by URL and show run results as they become available.
- **`cancel`** – Stop any future runs of a task.

© 2018, http://www.perfsonar.net

Diagnostics and Administrivia

- **`ping`** – Determine if pScheduler is running on a host.
- **`clock`** – Check and compare the clock(s) on pScheduler host(s).
- **`debug`** – Enable debugging on pScheduler's internal parts.
  - Only needed for debugging pScheduler itself.
- **`diags`** – Produce a diagnostic dump for the perfSONAR team to use in resolving problems.
- **`troubleshoot`** – make sure pscheduler is alive (runs ping, clock and more)

© 2018, http://www.perfsonar.net

perfSONAR

The `task` Command

Making Measurements

© 2018, http://www.perfsonar.net

The **`task`** Command

- Asks pScheduler to do some work

- Replaces the **`bwctl`** family of commands used in earlier versions of perfSONAR

© 2018, http://www.perfsonar.net

Synopsis

`pscheduler task [ task-opts ] test [ test-opts ]`

- ***`task-opts`*** – Switches related to everything but the test itself
  - Scheduling
  - Other behaviors (output format, etc.)
- ***`test`*** – What test the task is to perform (e.g., `throughput` or `trace`)
- ***`test-opts`*** – Test-specific switches and parameters

© 2018, http://www.perfsonar.net

Starting Simple

```
pscheduler                      Front-end command
    task                            pScheduler command
    rtt                             Test type (round-trip time)
    --dest localhost            Where the pings go
    --length 512                Packet size in bytes
```

Line breaks and indentation added for clarity.

© 2018, http://www.perfsonar.net

The Output Part I

```
% pscheduler task rtt --dest localhost --length 512
Submitting task...
Task URL:
https://ps.example.net/pscheduler/tasks/87e29f38-5b46…
Fetching first run...

Next run:
https://ps.example.net/pscheduler/tasks/87e29f38-5b46…
Starts 2016-12-07T07:57:30-05:00 (~7 seconds)
Ends   2016-12-07T07:57:41-05:00 (~10 seconds)
```


© 2018, http://www.perfsonar.net

The Output Part II

```
Waiting for result...

1    127.0.0.1  520 Bytes  TTL 64  RTT   0.0430 ms
2    127.0.0.1  520 Bytes  TTL 64  RTT   0.0590 ms
3    127.0.0.1  520 Bytes  TTL 64  RTT   0.0640 ms
4    127.0.0.1  520 Bytes  TTL 64  RTT   0.0540 ms
5    127.0.0.1  520 Bytes  TTL 64  RTT   0.0620 ms

0% Packet Loss  RTT Min/Mean/Max/StdDev =
0.043000/0.056000/0.064000/0.010000 ms

No further runs scheduled.
```


© 2018, http://www.perfsonar.net

Specifying Durations

- Subset of ISO 8601 Duration:
  - `PT19S` — *19 seconds*
  - `PT3M` — *3 minutes*
  - `PT2H5M` — *2 hours, 5 minutes*
  - `P1D` — *1 day*
  - `P3DT2H46M` — *3 days, 2 hours, 46 minutes*
  - `P2W` — *2 weeks*
- Inexact units (months, years) are not supported.

© 2018, http://www.perfsonar.net

Specifying Dates and Times

- ISO 8601 timestamp:
  - Absolute `2016-03-19T12:05:19`

- Coming in a future release:
  - Relative to Now `PT10M` *ISO 8601*
  - Even Boundary `@PT1H` *@ + ISO 8601 Duration*

© 2018, http://www.perfsonar.net

Task Options: Start Time

- **`--start t`** – Start at time *t*.
- **`--slip d`** – Allow the start time of run(s) to slip by duration *d*.
- **`--randslip f`** – Randomize slip time as fraction *f* of available. (Range `[0.0, 1.0]`)

© 2018, http://www.perfsonar.net

Task Options: Start Time

```
pscheduler
      task
      --start 2017-05-01T12:00
      --slip PT8M
      --randslip 0.5
      rtt
      --dest www.example.com
```

Start May 1, 2017 at noon

Slip start up to 8 minutes

Randomly slip up to 4 minutes

Line breaks and indentation added for clarity.

© 2018, http://www.perfsonar.net

Task Options: Repetition

- **`--repeat d`** – Repeat runs every duration *d*.
  - Other forms (notably CRON-like specification) to be added later.
- **`--until t`** – Continue repeating until time *t*.
  - Default is forever.
- **`--max-runs n`** – Allow the task to run up to *n* times.
  - Default is no upper limit.

© 2018, http://www.perfsonar.net

Task Options: Selecting a Tool

- **`--tool t`** – Add tool t to the list of tools which can be used to run the test.
  - Can be specified multiple times for multiple tools.

- If not provided, a tool is automatically selected from those available.

© 2018, http://www.perfsonar.net

Test Options

- Parameters for the test
  - Dependent on which test is being carried out.
  - See guide documents for each test for specifics.

- Example:

```
psc task … trace --dest host.example.org
```


© 2018, http://www.perfsonar.net

Putting the Parts Together

```
psc task
      --start 2016-05-04T19:20
      --repeat PT15M
      --max-runs 100
      trace
      --dest ps.example.org
      --length 384
      --hops 42
```

Start at the specified time

Repeat every 15 minutes

Stop after 100 successful runs

Trace to `ps.example.org`

Send 384-byte packets

Max. 42 hops to the destination

Line breaks and indentation added for clarity.

© 2018, http://www.perfsonar.net

Other Useful pScheduler Commands

`$ pscheduler plugins tests` *(Or* **`tools`** *or* **`archivers`***.)*

List all tests/tools/archivers available on the server

`$ pscheduler task clock --source host1 --dest host2`

Measure the clock difference between two hosts

`$ pscheduler task dns --query www.es.net --record a`

Measure the time to do a DNS lookup

`$ pscheduler schedule --filter-test=throughput`

Show the upcoming throughput tests

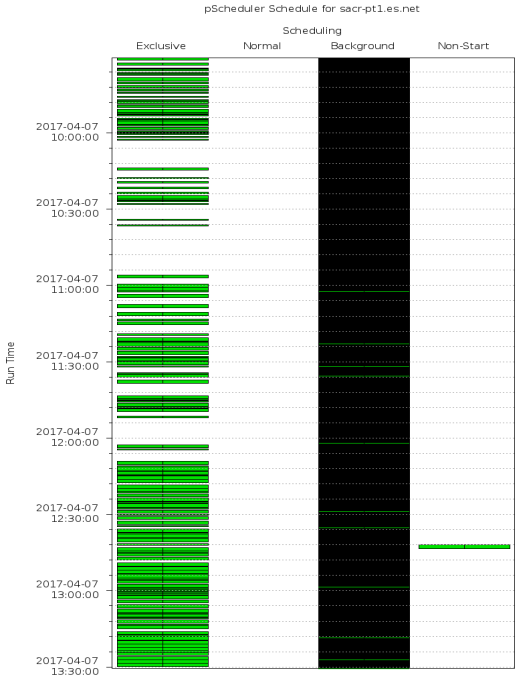
`$ pscheduler schedule --filter-test=throughput -PT1H --host somehost`

Show the throughput tests run in the past hour on ***somehost***

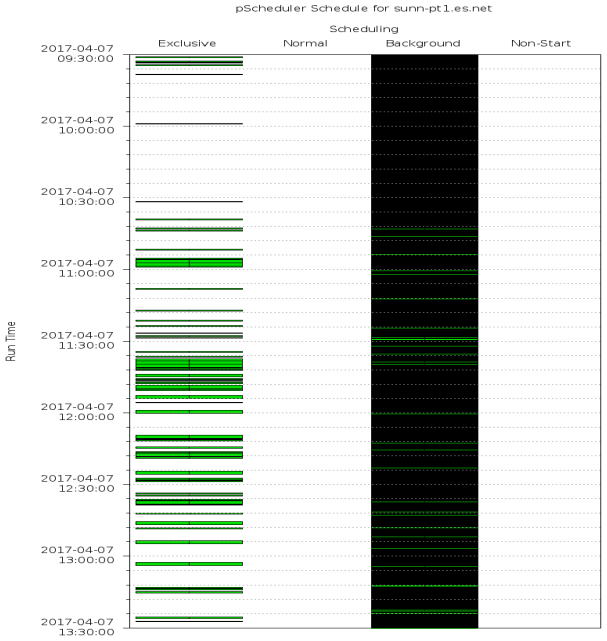
© 2018, http://www.perfsonar.net

Plotting the Schedule

```
$ pscheduler plot-schedule -PT2H > plot.png
```



From these plots, decided to move some tests from sacr-pt1.es.net to sunn-pt1.es.net



© 2018, http://www.perfsonar.net

Sample pScheduler Throughput Command

- **Old:**

```
$ bwctl -c receive_host -s send_host -t 30
```

- **New:**

```
$ pscheduler task throughput
  --source send_host
  --dest receive_host
  --duration PT30S
```

- For more details on commands see http://docs.perfsonar.net/pscheduler_intro.html

© 2018, http://www.perfsonar.net

Sample pScheduler Packet Loss/Latency Test Command

- **Old:**

```
$ bwping -s send_host -c receive_host
$ bwping -T owamp -s send_host -c rcv_host -N 1000 -i .01
```

- **New:**

```
$ pscheduler task rtt --source send_host --dest rcv_host
$ pscheduler task latency
  --source send_host
  --dest receive_ host
  --packet-count 1000
  --packet-interval .01
```


© 2018, http://www.perfsonar.net

Sample pScheduler Traceroute Command

- **Old:**

```
$ bwtraceroute -c receive_host -s send_host
```

- **New:**

```
$ pscheduler task trace
  --source send_host
  --dest receive_host
```


© 2018, http://www.perfsonar.net