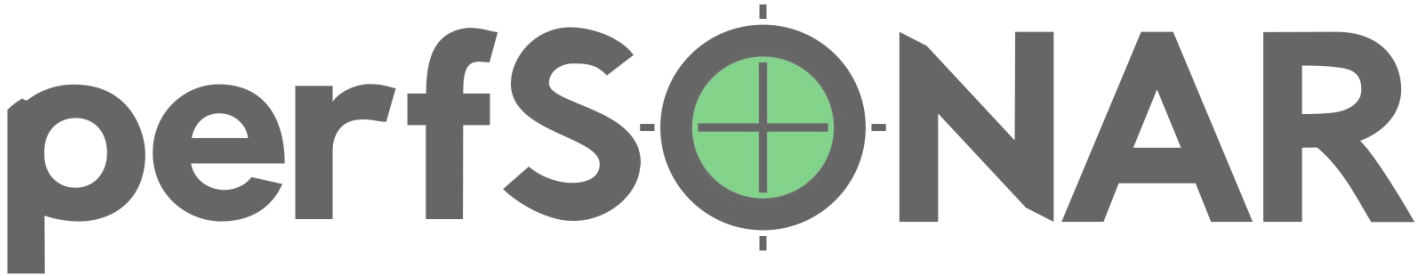


pScheduler

The perfSONAR Scheduler

Network Performance and Monitoring workshop

Lætitia Delvaux, PSNC, laetitia.delvaux@man.poznan.pl
Krešimir Šodan, CARNET, kresimir.sodan@carnet.hr

Outline

- What is pScheduler?
- pScheduler CLI basics
- The task command
- Other test types
- Hands-On
  - On your perfSONAR host

What is pScheduler?

- Software for scheduling, supervising and archiving measurements.
- Client / Server architecture
  - Most of the hard work done by a well-proven RDBMS (PostgreSQL)
- REST API
- Standardized, documented data formats using JavaScript Object Notation (JSON)

Extensible Architecture

- Plug-in system allows integration of new…
  - Tests — *Things to measure*
  - Tools — *Things to do the measurements*
  - Archivers — *Ways to dispose of results*
  - Contexts — *Environment where tests are run*

- 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

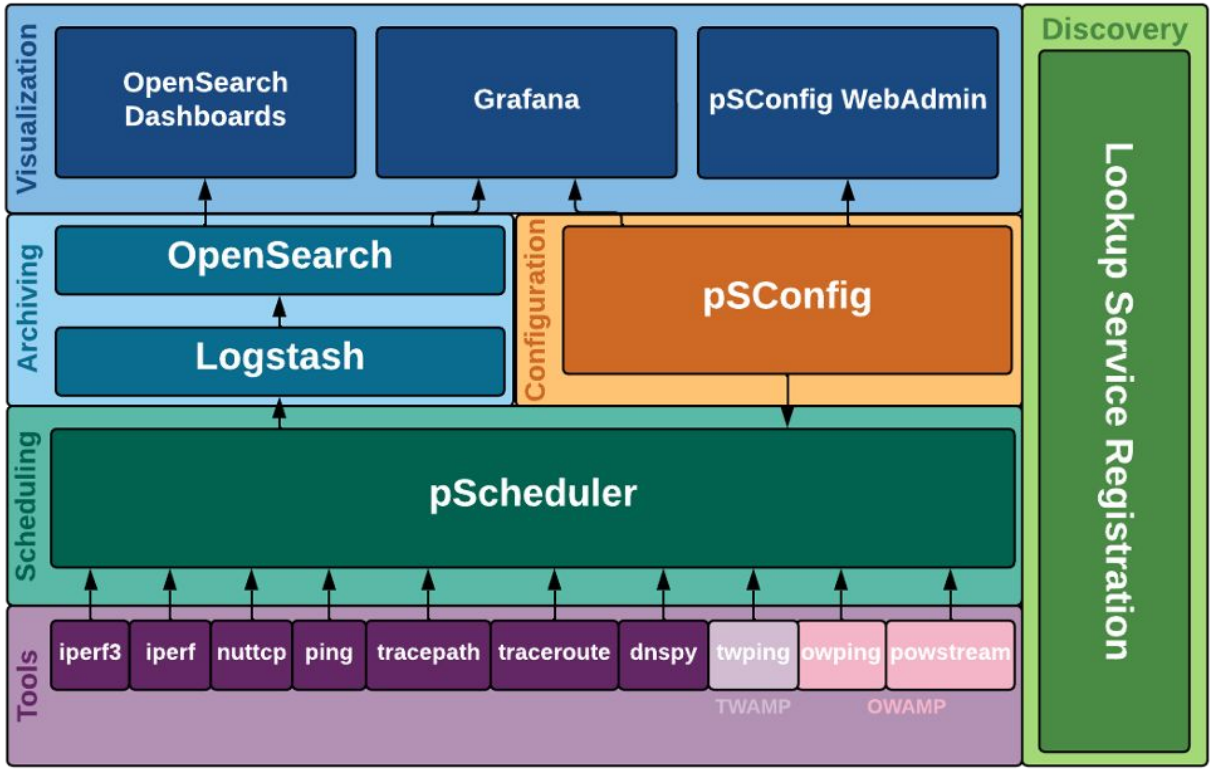
Test Abstraction

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

pScheduler Archivers

- Support for
  - HTTPS GET/PUT (OpenSearch etc.)
  - RabbitMQ
  - Kafka
  - Syslog
- Like tools and tests, archivers are pluggable
  - Well-defined API
  - Easy to add additional archive targets
- Archiving is reliable to reduce data loss during failures

pScheduler in the perfSONAR architecture



pScheduler CLI Basics

Front End

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

`pscheduler`

- Autocompletes easily on most systems:

`psch` *Tab*

Command Format

- All commands follow the same format:

pscheduler *command* [*arg* …]

Getting Help

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

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

Task Commands

- **`task`** – Give pScheduler a task that consists of making one or more measurements (*runs*).

- Secondary commands
  - **`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.

Diagnostics and Administrivia

- **`ping`** – Determine if pScheduler is running on a host.
- **`clock`** – Check and compare the clock(s) on pScheduler host(s).
- **`troubleshoot`** – make sure pscheduler is alive (runs ping, clock and more)
- **`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.

The `task` Command

Making Measurements

The `task` Command

- Asks pScheduler to do some work
- To make a measurement

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

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.

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)
```

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.
```

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.

Specifying Dates and Times

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

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

Task Options: Start Time

- **`--start t`** – Start at time *t*.

- **`--slip d`** – Allow the start time of run(s) to slip by duration *d*.
  - defaults to PT5M (or `PSCHEDULER_SLIP` from ENV)
- **`--sliprand`** – Randomly choose a timeslot within the allowed slip instead of choosing earliest available

Task Options: Start Time

```
pscheduler
    task
    --start 2017-05-01T12:00    Start May 1, 2017 at noon
    --slip PT8M                 Slip start up to 8 minutes
    --sliprand                  Random start within the 8 mins
    rtt
    --dest www.example.com
```

Line breaks and indentation added for clarity.

Task Options: Repetition

- **`--repeat d`** – Repeat runs every duration *d*.
  - Other forms (notably CRON-like specification) to be added later (4.3).
- **`--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.

Task Options: Behavior

- **`--import` *`f`*** – Import JSON for the task from file *`f`* (use – for standard input)
- **`--export`** – Dump the task specification as JSON to standard output but don't run it.
- **`--url`** – If the task is created, dump its URL to standard output and exit.
- **`--format` *`f`*** – If results are to be displayed, use format ***f***, which is one of `text` (the default), `html` or `json`.
- **`--assist` *`s`*** – Ask server *`s`* for assistance in setting up the task
  - Use this when the pScheduler server is not available on the local host.
  - `PSCHEDULER_ASSIST` from the environment

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.
  - Throughput: iperf3
  - Latency: owping
  - Trace: tracepath

Test Options

- Parameters for the test
  - Dependent on which test is being carried out.
  - See guide documents for each test for specifics (**`--help`**).

- Example:

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

Putting the Parts Together

```
pscheduler 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.

Other test types

And tools

Application level tests

- DNS
  - **`pscheduler task dns --query perfsonar.net`**
- HTTP
  - **`pscheduler task http --url https://docs.perfsonar.net`**
- Disk to disk (tools: **`curl ftp globus`**)
  - **`pscheduler task disk-to-disk --source ftp://speedtest.tele2.net/1KB.zip --dest /tmp/test.out`**
- And possibly more…
  - **`pscheduler plugins tests`** *(Or* **`tools`** *or* **`archivers`** *or* **`contexts`***)*

  List all tests/tools/archivers/contexts available on the server

Other Useful pScheduler Commands

$ pscheduler troubleshoot

$ pscheduler troubleshoot host2

Test pscheduler availability and readiness on local and remote host

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

Measure the clock difference between two hosts

$ pscheduler schedule --filter-test=throughput

Show the upcoming throughput tests (1 hour ahead)

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

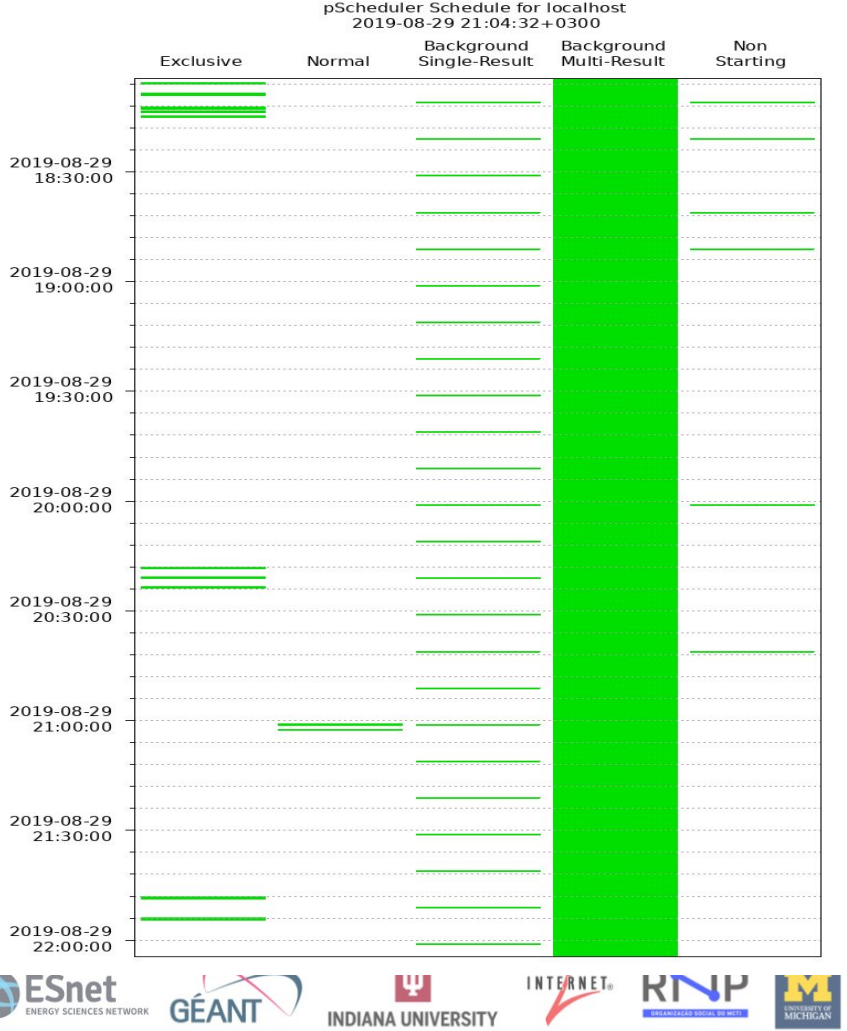
Show the throughput tests run in the past hour on ***somehost***

$ pscheduler monitor

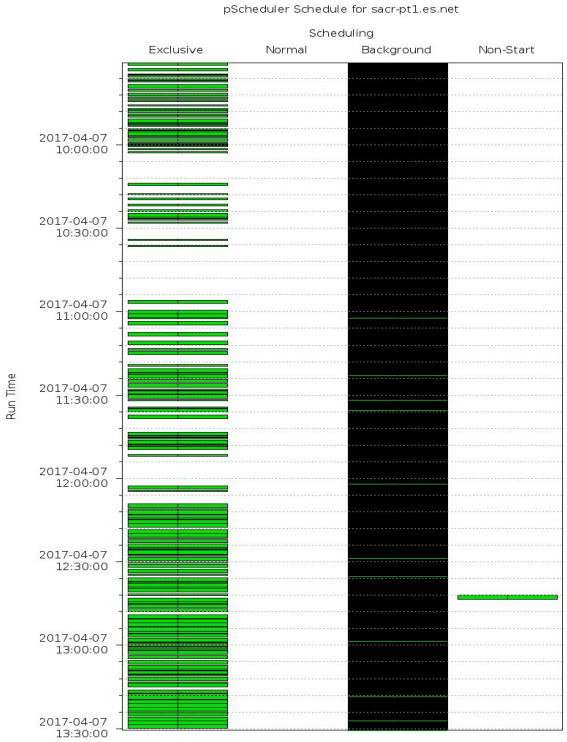
Live view on tasks

Plotting the Schedule

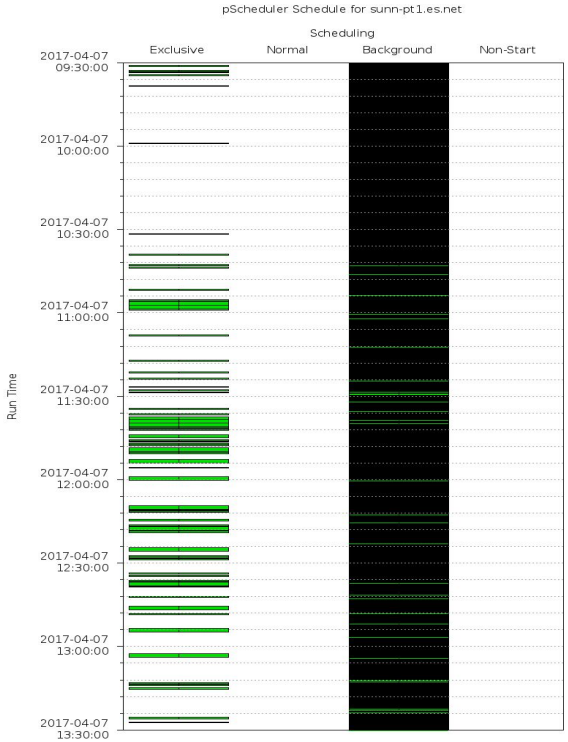
```
$ pscheduler plot-schedule -PT4H
    > /var/www/html/schedule.png
```



Comparing the schedules



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



Saving a Task for Re-Use

```
pscheduler task
    --repeat PT15M
    --max-runs 100
    --export
    trace
    --dest ps.example.org
    --length 384
    --hops 42
    > mytask.json
```

Repeat every 15 minutes

Stop after 100 successful runs

*Dump task JSON to standard output**

Trace to `ps.example.org`

Send 384-byte packets

Max. 42 hops to the destination

Redirect output to `mytask.json`

* Exporting the task means pScheduler does not run it.

Re-Using a Saved Task

```
pscheduler task
    --import mytask.json   Import contents of mytask.json
    --repeat PT10M         Repeat every 10 minutes*
    --max-runs 20          Stop after 20 successful runs*
    trace
    --dest ps.example.edu  Trace to ps.example.edu *
```

*These switches augment or change values specified in `mytask.json`.

Line breaks and indentation added for clarity.

Clever Trick: Editing Task Files

```
pscheduler task
    --import mytask.json        Import contents of mytask.json
    --repeat PT1H30M            Repeat every 90 minutes*
    --max-runs 68               Stop after 68 successful runs*
    --export                    Dump task JSON to standard output†
    trace
    --dest ps.example.edu       Trace to ps.example.edu *
    > othertask.json            Redirect output to othertask.json
```

* These switches augment or change values specified in `mytask.json`.

† Exporting the task means pScheduler does not run it.

Line breaks and indentation added for clarity.

Administrative Commands

End Users: Here Be Dragons

The **debug** Command

`psc debug` ***on|off*** `[` ***service*** `…]`

- Enables or disables debugging in various pScheduler services on the local system.
- Output goes to syslog. Configure syslogd for `*.debug`.
- Must be run as `root` or `pscheduler`.*
- Services: (Default is to operate on ***all*** services.)

```
ticker      runner          api
scheduler      archiver
```

- **This command is for debugging pScheduler, not for test diagnostics.**

*May change to `perfsonar` in the future.

pScheduler Hands-On

Some more examples…

Ping, troubleshoot and task

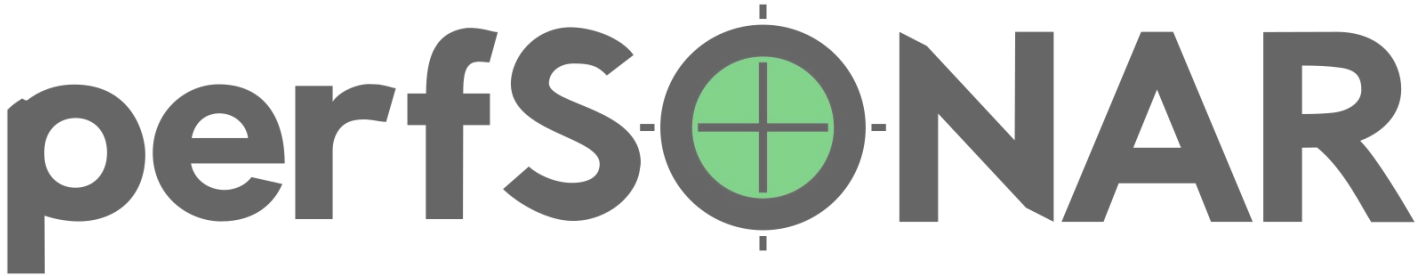
- Run the pscheduler ping command
  - pscheduler ping <ipAddress>
- Run the pscheduler troubleshoot command to another MP
  - pscheduler troubleshoot <ipAddress>
- Run the pscheduler clock command
  - pscheduler task clock --dest <ipAddress>
- Run throughput test to another MP
  - pscheduler task throughput --dest <ipAddress>
  - pscheduler task throughput --source <ipAddress> --dest <ipAddress> –u
  - pscheduler task --slip PT20M --sliprand throughput --dest <ipAddress>
- Run latency test to another MP
  - pscheduler task latency --dest <ipAddress>
  - pscheduler task latency --source <ipAddress> --dest <ipAddress> --ip-version 6

Advanced tasks, getting help and limits

- Debugging
  - pscheduler task –debug latency --source <ipAddress> --dest <ipAddress>
- Getting help
  - pscheduler
  - pscheduler task --help
  - pscheduler task latency --help
- UDP throughput limits
  - pscheduler task throughput --source <ipAddress> --dest <ipAddress> -u -b 200M
  - Limited to 50M because of /etc/pscheduler/limits.conf

Checking the schedule

- List
  - pscheduler schedule +PT1H | less
- Live
  - pscheduler monitor
- Plot
  - sudo mkdir –p /var/www/html/workshop
  - sudo chgrp pssudo /var/www/html/workshop
  - sudo chmod g+w /var/www/html/workshop
  - pscheduler plot-schedule > /var/www/html/workshop/schedule.png
  - https://<ipAddress>/workshop/schedule.png



pScheduler

The perfSONAR Scheduler

Network Performance and Monitoring workshop

Lætitia Delvaux, PSNC, laetitia.delvaux@man.poznan.pl
Krešimir Šodan, CARNET, kresimir.sodan@carnet.hr