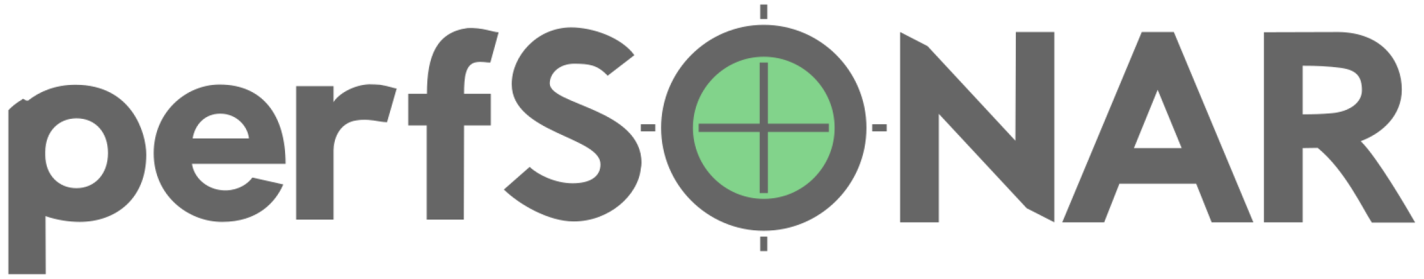


Plugging In

Bringing New Measurements to perfSONAR

Mark Feit ▪ Internet2 / The perfSONAR Development Team ▪ mfeit@internet2.edu

Second European perfSONAR Workshop

perfSONAR is developed by a partnership of

pScheduler Plugins

- **Test** Abstract measurement
- **Tool** Carries out a test
- **Archiver** Disposes of results
- **Context** Changes tool's environment

What can perfSONAR measure?

- The usual suspects:
  - Throughput
  - Latency
  - Round-trip time
  - Path

What can perfSONAR measure?

- The less-usual suspects:
  - DNS and HTTP response time
  - Network reachability
  - S3 throughput
  - SNMP values
- These ship with the current release.

What can perfSONAR measure?

- The unusual suspects:
  - Path MTU
  - VoIP call quality
  - Interface traffic dump

- These don't exist yet.

What can perfSONAR measure?

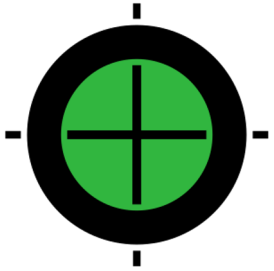
- Way out there:
  - Stock prices
  - Weather
  - Garage door state
- These are thought exercises.
  - No plans to do them.
  - But we could.
  - If provoked.

The pScheduler Magic Formula

Test + Tool = Measurement

Requirements for Tests and Tools

- **Test:** The measurement and its results must be describable in a concrete way.
- **Tool:** There must be a program to do the measurement.
- **Tool:** That program must be usable by another program.
- **Tool:** The results must be readable by a program in the tool plugin.



Test Design

Test Design Philosophy

- Focus on what's being measured, not the tool(s) doing the measuring.
- Survey the tool landscape and use tool capabilities to inform the design
- Don't make the test a proxy for a single tool
  - Hampers future flexibility

The Garage Door Test

- Determine the state of a garage door
  - How far along is it in its travel?
  - Is it moving?
  - Which way?

- Garage door controllers speak several different protocols
  - DoorML
  - MegaDoor 5000

- If the door is moving, some controllers don't give an answer until it stops.

The Garage Door Test Specification

Test Parameter	Description	Sample Value(s)
`door`	Door address	`dock17.chi6.example.com`
`protocol`	Garage door protocol	`doorml, megadoor5000`
`timeout`	How long to wait	`PT10S, (None)`

Needed to calculate time on the schedule.

Enables automatic selection of tools that speak one or more of the protocols.

- Standard test specification for all garage door tests.

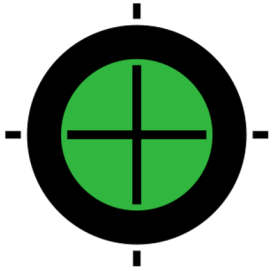
The ~~Lowest~~ Highest Common Denominator

- Test parameters and values don't have to be supported by all tools.
- Tool plugins…
  - …declare the tests they're willing to run.
  - …give a yes/no answer about ability to run a given specification.
- pScheduler will shop a test around to the tools to find candidates.
  - All tools that understand the test (automatic selection)
  - First in the specified list
- Those saying yes are sorted by preference order and one is selected.

The Garage Door Test Result

Result Parameter	Description	Sample Value(s)
`position`	How far open the door is as a fraction: • `0.0` = Fully closed • `1.0` = Fully open	`0.0, 0.3475, 1.0`
`direction`	Which way the door is moving: • `-1` = Toward closed • `0` = Not moving • `1` = Toward open	`-1, 0, 1`

- Standard result for all garage door tests.



Plugin Mechanics

General Plugin Structure

- Directory containing a standard set of *methods*.

- Each method serves a purpose.
  - Validate input
  - Determine if a measurement is possible
  - Carry out a measurement

- Each method is an executable program that…
  - …reads standardized JSON* data on standard input.
  - …writes standardized JSON* data on standard output.
  - …exits **0** normally or **1** in the event of a catastrophic failure.

  *Some of that JSON is free-form, i.e., any JSON is considered valid

The Process Boundary

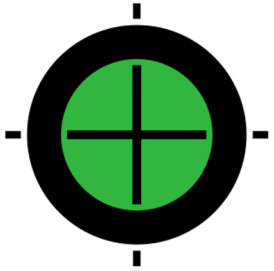
- Allows plugin methods to be written in any language
  - We use and recommend Python 3
  - Python **`pscheduler`** module available with classes and functions that do many of the operations plugins have in common
- Makes development, test and debug easier
  - Run methods individually on the command line with canned JSON input
  - No need to deal with the rest of pScheduler
- Protects pScheduler from plugin failures

Test Plugin Methods

`enumerate`	Describe the test plugin to pScheduler
`cli-to-spec`	Convert CLI switches to JSON
`spec-to-cli`	Convert JSON to CLI switches
`participants`	Determine pScheduler nodes involved
`spec-is-valid`	Determine if a test specification is valid
`spec-format`	Convert a test spec to text or HTML
`result-format`	Convert a test result to text or HTML
`limit-is-valid`	*Deprecated in favor of jq*
`limit-passes`	*Deprecated in favor of jq*

Tool Plugin Methods

• **`enumerate`**	Describe the tool plugin to pScheduler
• **`can-run`**	Determine if the tool can run a test
• **`duration`**	Calculate the time to run a test
• **`participant-data`**	Generate internal info for participants
• **`run`**	Measure and produce interim result
• **`merged-results`**	Combine interim results



Getting Your Plugins Developed

Help From the Development Team

- No comprehensive plugin development manual yet.
- Advice on test design
- Assistance with implementation or debugging
- Plugins of general interest may be adopted and become part of the standard perfSONAR distribution.

The Plugin Development Kit (PDK)

- Contains templates for each type of plugin
- Generates a skeleton in the pScheduler source tree
- Follow-the-numbers implementation

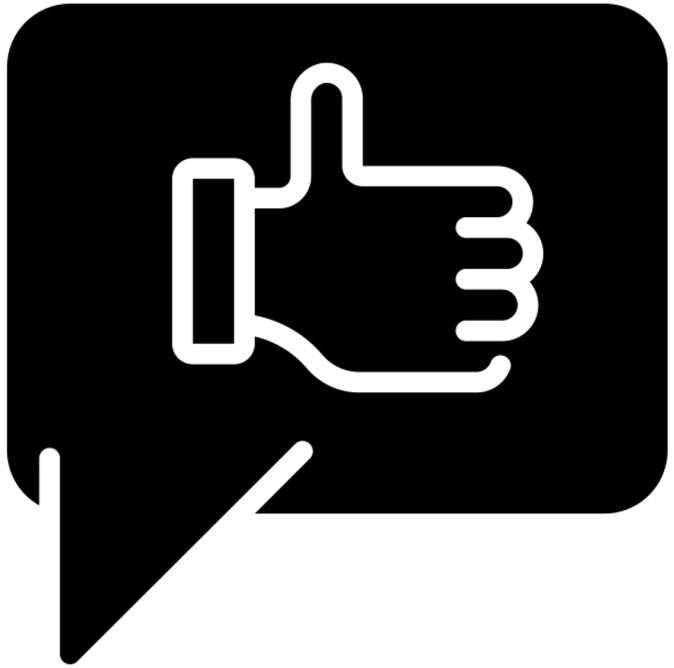
- Works on a standard pScheduler development cluster
- Easy with Vagrant (see **`scripts/vagrant/dev-cluster`**)

Question and answer icon by iconosphere from The Noun Project

Questions and Answers

mfeit@internet2.edu

perfSONAR



Thanks!

For more information,
please visit our web site:
https://www.perfsonar.net

Thanks icon by priyanka from The Noun Project

perfSONAR is developed by a partnership of ESnet GÉANT INDIANA UNIVERSITY INTERNET2 RNP ORGANIZAÇÃO SOCIAL DO MCTI UNIVERSITY OF MICHIGAN