perfSONAR

A New Architecture for Streaming Measurements with pScheduler

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

Third European perfSONAR User Workshop · May, 2022

perfSONAR is developed by a partnership of

©2022 The perfSONAR Project and its Contributors · Licensed CC BY-SA 4.0 · https://www.perfsonar.net

Advanced Material

- Material covered in this presentation is not necessary for everyday use of perfSONAR.

- This is pScheduler “inside baseball.”

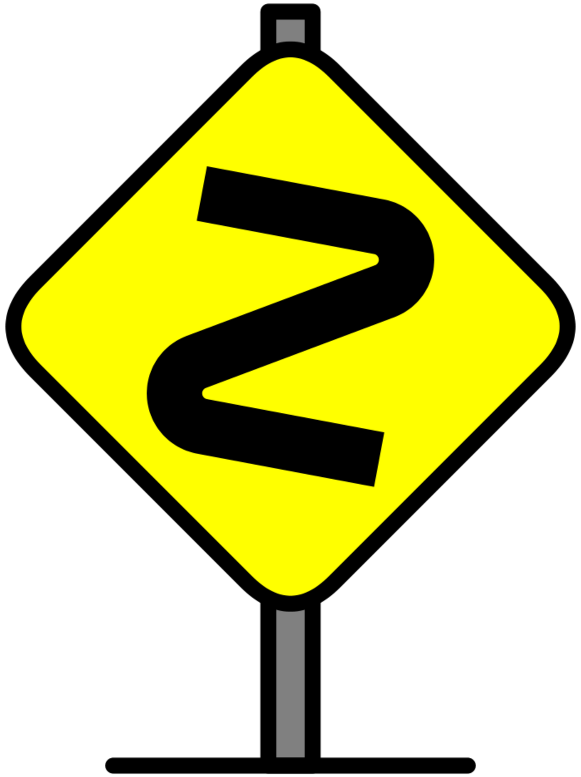


Image: MetaNest, CC-BY-SA 3.0

Disclaimer

Features described in this talk are being considered for a release that may happen sometime later than today.

None of it exists… *yet*.

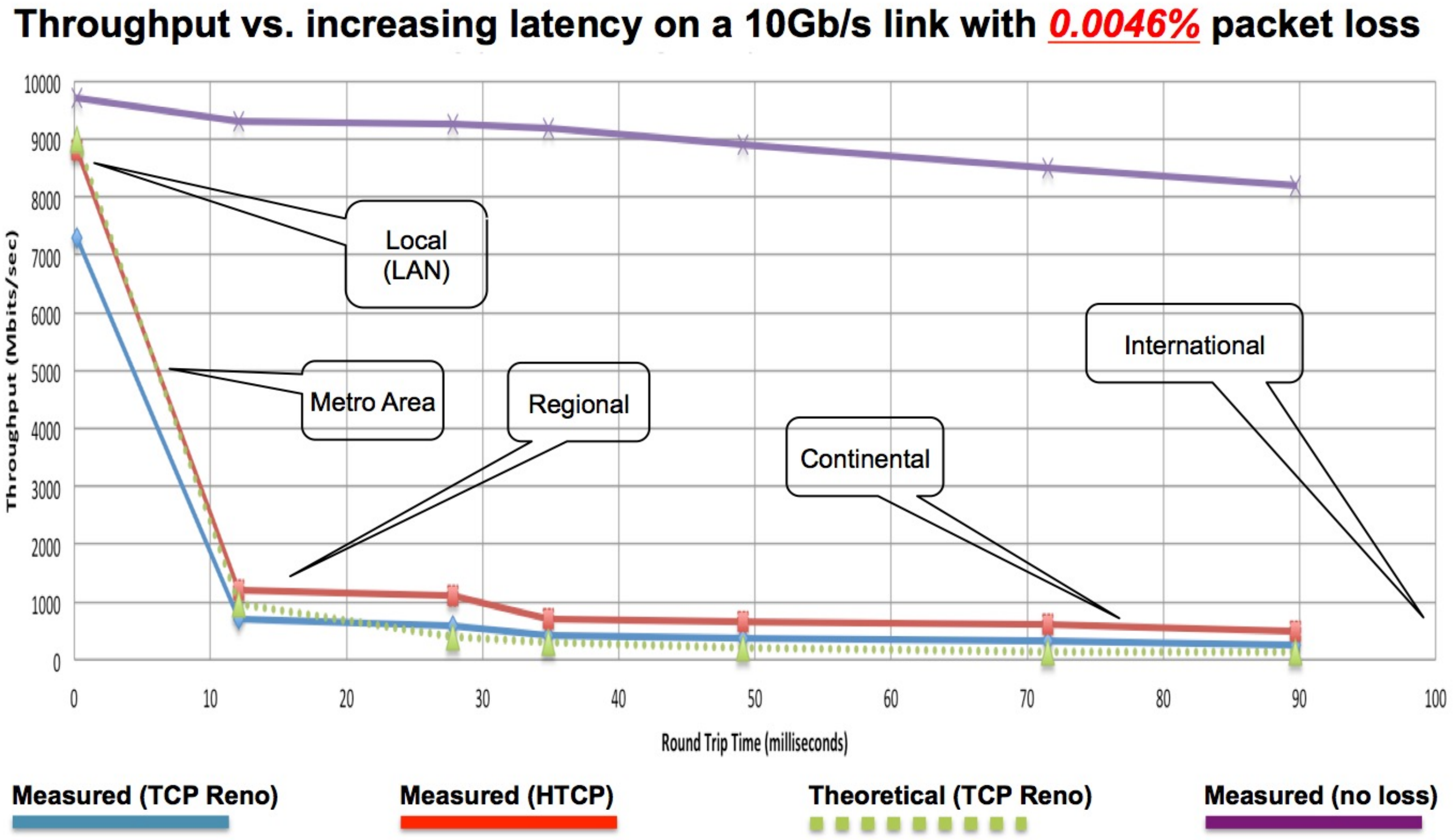
Streaming Measurements

- Some problem-causing events are transient.
  - Continuous throughput is expensive
  - Is the network there for test or user traffic?

- Some measurements can be done continuously.
  - Latency and loss are low-bandwidth

Hint, Hint: Implied Problems

- Packet loss on longer links means loss of throughput on TCP streams.
- Is this throughput measurement really necessary?
- Probably not. Find and fix the loss.



Single-Measurement Resource Consumption

Thread	pScheduler Runner service
Process	pScheduler tool plugin **`run`** method
Process	Measurement tool (**`ping`**, **`iperf3`**)

Powstream

- Part of the OWAMP family
- Continuous measurements (Latency / Loss /Jitter)
- Aggregates multiple measurements into a single result
    - Optional per-packet data

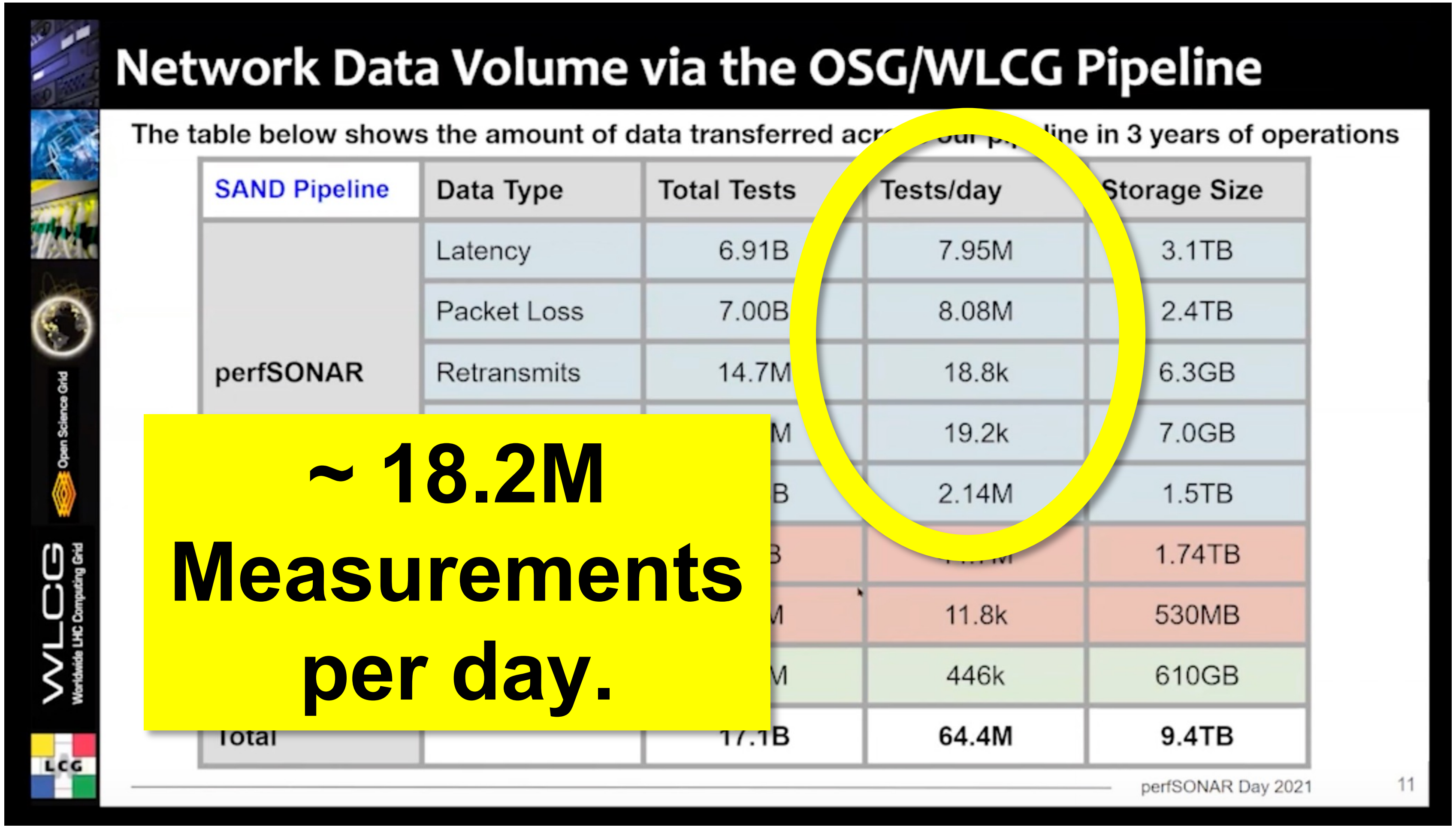
No Such Thing as a Free ~~Lunch~~ Measurement

- Running Powstream consumes more resources:
  - Two processes to conduct the measurement.
  - Process run periodically by tool plugin to convert results into something usable.
  - Total: Thread + 4 processes + Itinerant Process

It Sounds Worse Than it Is

- Many copies of the same programs running at the same time
- Shared code and data pages

Large-Scale Applications



Network Data Volume via the OSG/WLCG Pipeline

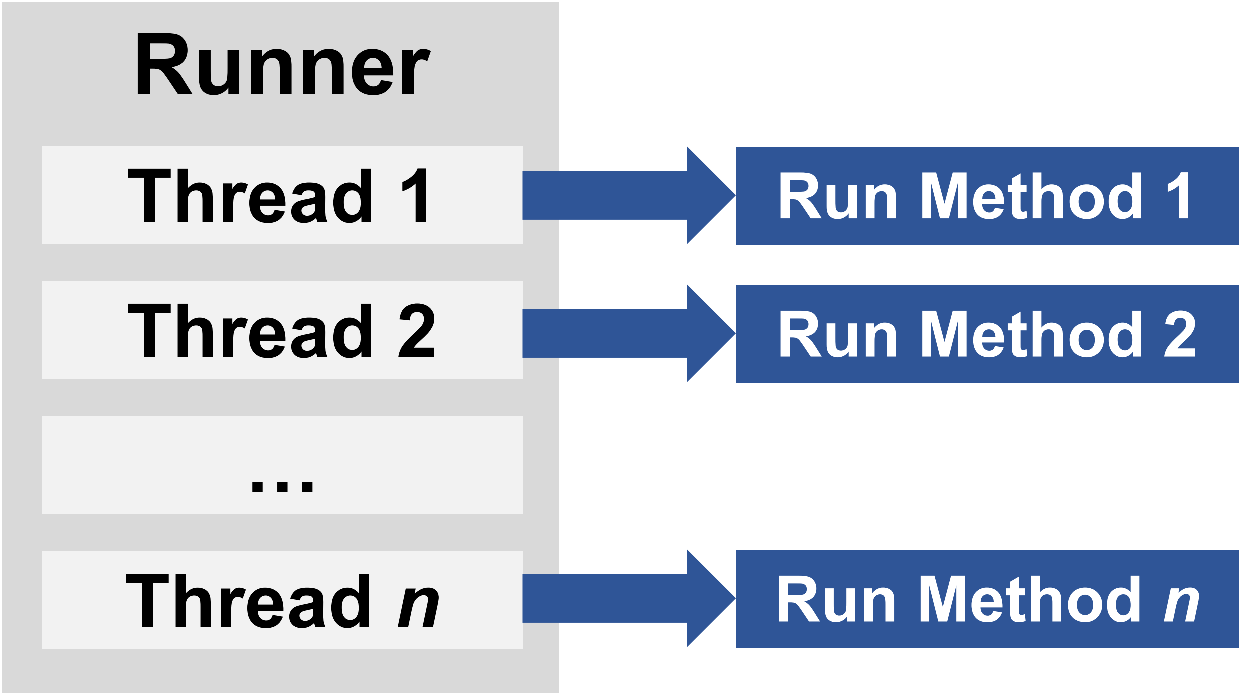
The table below shows the amount of data transferred across our pipeline in 3 years of operations

SAND Pipeline	Data Type	Total Tests	Tests/day	Storage Size
perfSONAR	Latency	6.91B	7.95M	3.1TB
	Packet Loss	7.00B	8.08M	2.4TB
	Retransmits	14.7M	18.8k	6.3GB
[illegible]	[illegible]	M	19.2k	7.0GB
[illegible]	[illegible]	B	2.14M	1.5TB
[illegible]	[illegible]	B	[illegible]	1.74TB
[illegible]	[illegible]	M	11.8k	530MB
[illegible]	[illegible]	M	446k	610GB
Total		17.1B	64.4M	9.4TB

~ 18.2M Measurements per day.

From Shawn McKee’s 2021 perfSONAR Day presentation.

4.*x* Threading Architecture



Strangled by the Python

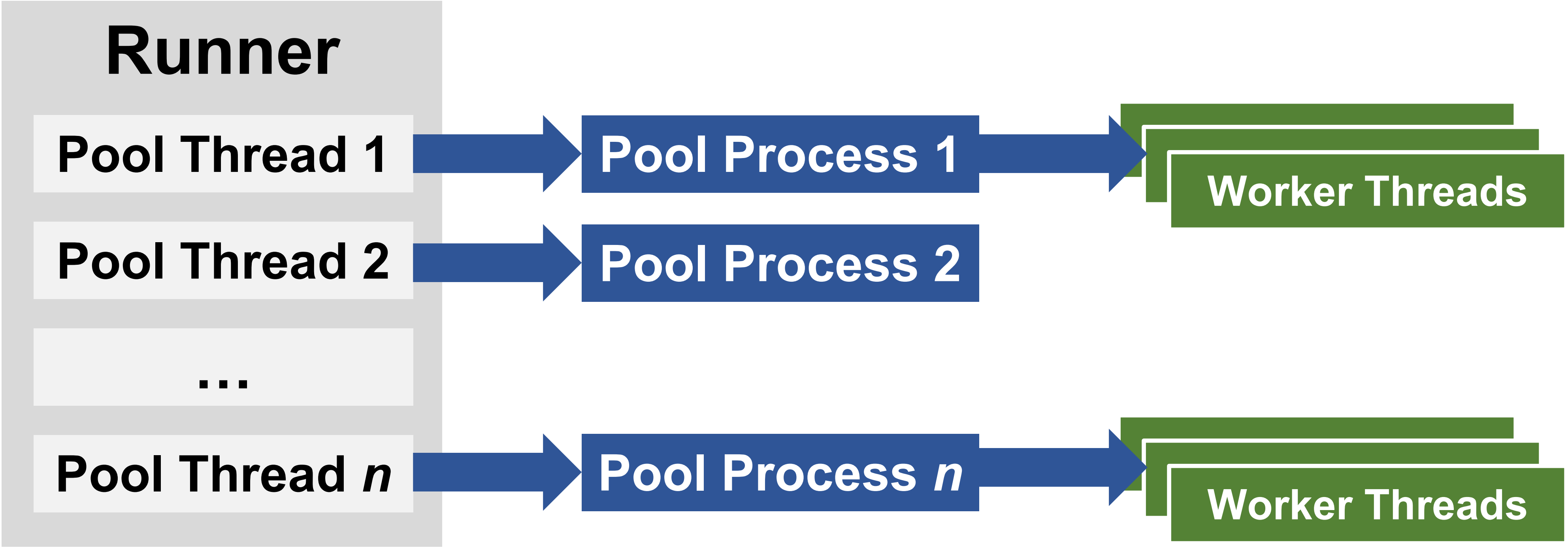
- Python was selected for pScheduler because it's well-understood within the user community.

- It has threads but is effectively single-core because of the Global Interpreter Lock (GIL).

New Threading Architecture in 5.0

- The GIL limits the number of usable threads.
- Work delegated to child processes
- Relatively-small number of threads per child. (20)
- Takes better advantage of more cores when available.

New Threading Architecture



Runner

Pool Thread 1

Pool Thread 2

…

Pool Thread *n*

Pool Process 1

Pool Process 2

Pool Process *n*

Worker Threads

Worker Threads

Pool Process Management

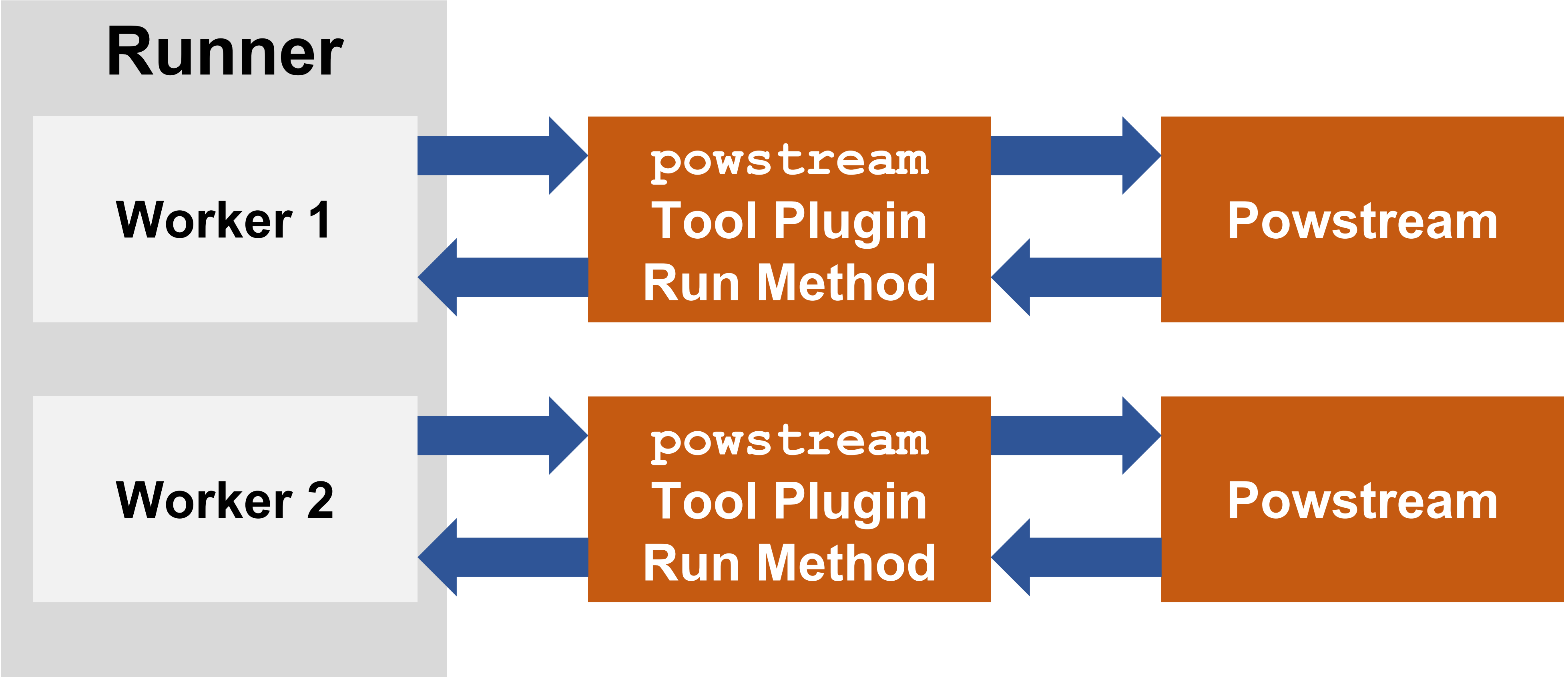
- Pool processes create worker threads per job.
- Distribution of jobs favors a lower number of pool processes.
- Idle processes go away.
- Pools can have a limited lifetime
  - E.g., 10,000 jobs and that’s it
  - Prevents problems caused by memory leaks

Solving the Powstream Problem

How’s that again?

- Resource consumption
- New applications that want a real-time stream of individual measurements
  - One-minute, aggregated granularity with optional individual packet data isn’t good enough.
- Powstream was never designed with either in mind.

Conventional Measurement



New Concept: Unsupervised Measurements

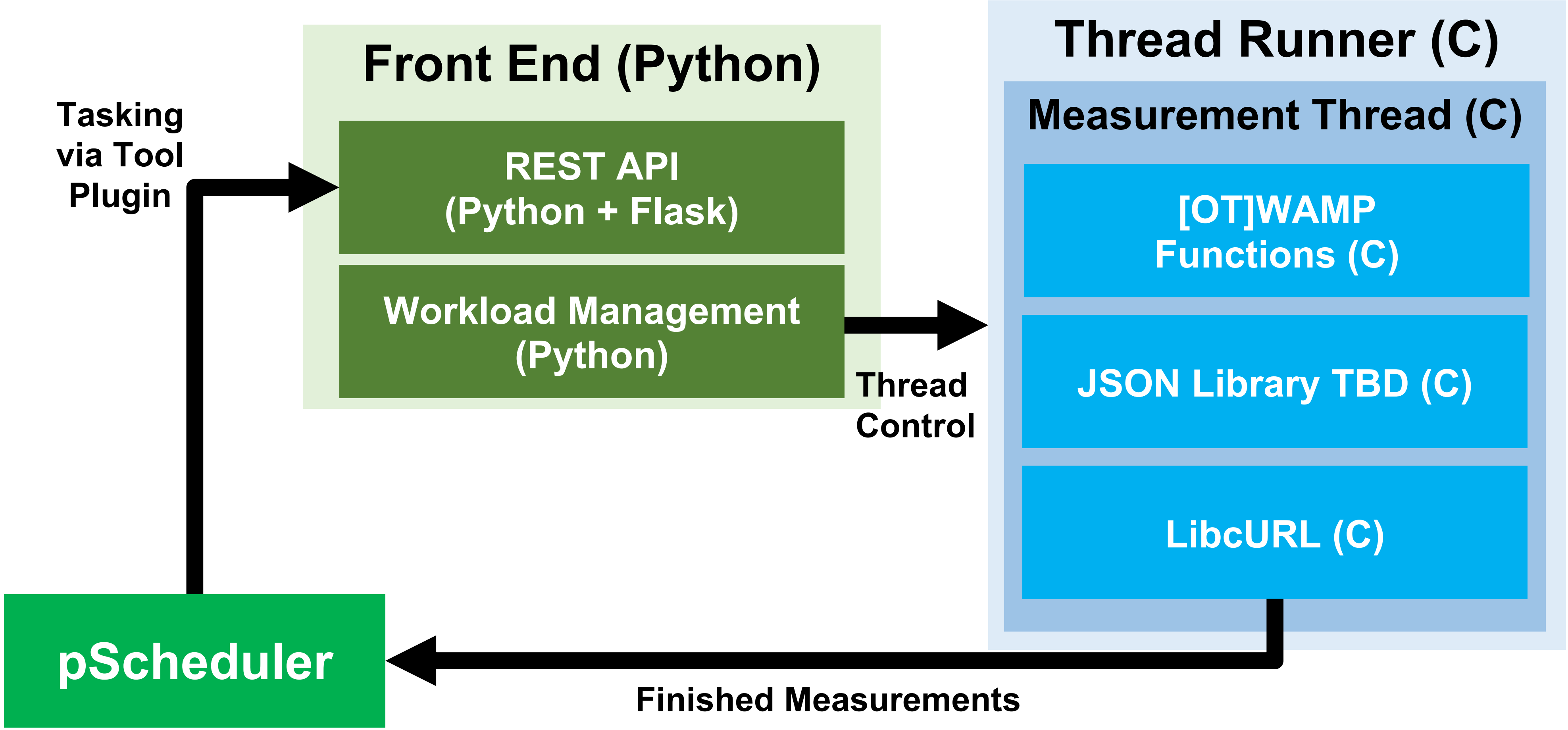
- Variation on tool plugin. Runs measurements independently.
- New **`start`** method establishes a long-term, multi-result measurement with an external service.
  - Provides information about where to post results
  - Authentication key
- Service sends results directly into pScheduler via the API.
- Lacks conventional measurement’s persistent **`run`** method.
- New **`check`** method in plugin called to check the measurement
  - Re-establishes if not.

pSlam: pS Latency Measurement Service

- Takes the place of Powstream
- Does measurements as directed

- Architecture takes advantage of better threading
- Avoids Python’s pitfalls

pSlam: pS Latency Measurement Service



pSlam: How do we get there?

- Isolate OWAMP/TWAMP measurement functions from the reference implementations.
- Make them callable as utilities
- Change pScheduler support unsupervised measurements
- Develop measurement thread and thread runner
- Develop front end
- Develop pScheduler tool plugin
- Retire Powstream

When?

- Most of this talk encompasses the basic design.
- Isolation of [OT]WAMP functions is already underway at ESnet.
- Development of everything else starts this summer.
- Look for this in 5.1 or 5.2.
    - Other fish to fry

More Disclaimers

- pScheduler is not suitable for every streaming application

- 5.0 will be better at handling high volumes than 4.*x*.
  - We don’t know how much better yet.

- Direct-to-archive makes sense in some cases:
  - Very-high volume
  - Ultra-low latency demands
  - No need for pScheduler’s post-processing or archive flexibility

perfSONAR

Thanks icon by priyanka from The Noun Project

Thanks!

Email:
mfeit@internet2.edu

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

perfSONAR is developed by a partnership of

ESnet GÉANT INDIANA UNIVERSITY INTERNET2 UNIVERSITY OF MICHIGAN

©2020 The perfSONAR Project and its Contributors · Licensed CC BY-SA 4.0 · https://www.perfsonar.net