Codebase stories: A year in the Parsl git repo

Ben Clifford
benc@hawaga.org.uk
ParslFest 2024

418 PRs merged to master

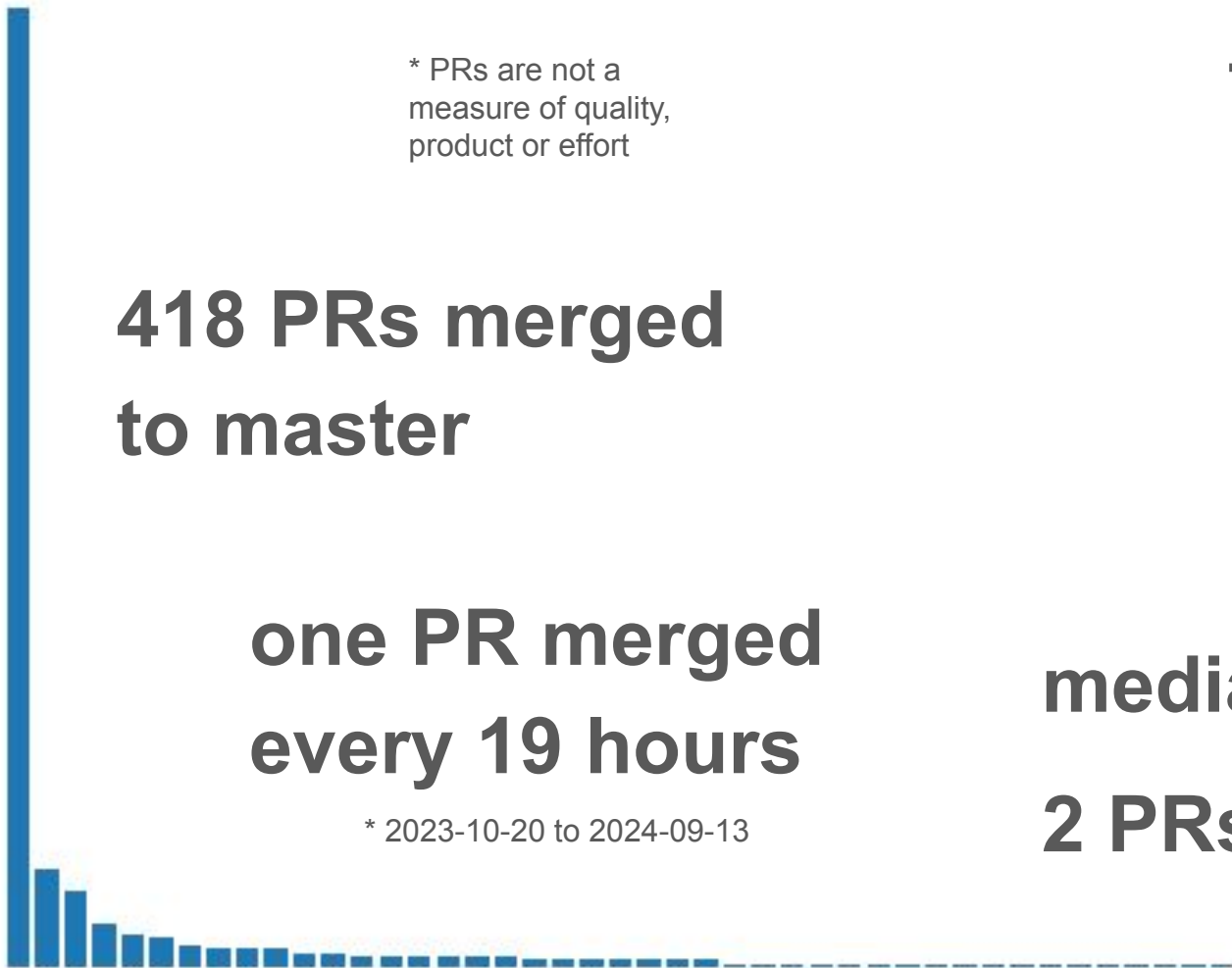* PRs are not a measure of quality, product or effort

from 43 people

* Pull requests to main codebase, not broader contributions

one PR merged every 19 hours

* 2023-10-20 to 2024-09-13

median 2 PRs/committer

Story 1: Tidyup of block scaling code

the code that starts
and stops blocks
(batch jobs)

what if we rearranged this code so
these bugs were easy to fix?

shared between Parsl
and Globus Compute

me + Kevin: many small PRs
(eg rename this variable, move
this function)
PRs shouldn't change behaviour -
no "fixing bugs by the way"

several layers of "research
prototyping"
=> 2 bugs that were hard to fix

outcome:
2 bugs were easy to fix! (almost
one-liners)
uncovered a new integration bug
(also a 1 liner to fix!)

Story 2: Outreachy.org

"Outreachy provides **internships in open source and open science.** Outreachy provides internships to people subject to systemic bias and impacted by underrepresentation in the technical industry where they are living."

non-traditional application process: applicants make contributions to projects

long tail of 1/2 contributions is heavily outreachy

🤯

selected 2 applicants for 2 projects

1 funded: Kelechi
http://parsl-project.org/2024/07/31/Meet-Kelechi-Nwankwo.html

I made many more "starter" issues to help people find things to do

Story 3: No more channels

Channels are deprecated, removal in October.
https://github.com/Parsl/parsl/issues/3515
Very few users
I was hoping for usage info to help me justify removal

⭐ improve or delete 💀

You can still run workers remotely.
But Parsl gives you less help now.

Channels: a half-hearted attempt to do remote execution

Remote execution = submitting batch jobs over SSH

Ignores:
- (no) shared filesystem
- network connectivity for pilot workers (NAT/firewall)
- consistent Python environments
- bad WAN-style networking
- authentication mechanisms

More attention to Globus Compute <-> Parsl [* Yadu]

Story 4: Shut down and exit reliability

See the safe-exit tag in github

things being left behind   hangs at exits

Python atexit handlers
Process shutdown
Garbage collectors
Race conditions   `with parsl.load(config):`
Unix signals   `# your workflow here`
Interchange RPC
Protocol reliability assumptions
Who scales which batch jobs?
The fundamental wrongness of
fork-without-exec

Story 5: Language features

language club vs task club
"how to describe tasks" vs "how to execute tasks"

dependency resolution on structures:

slow trend towards using
more Pythonic language
features but making them
somehow Parsl-ish

```
v = [future1, future2, future3]

future4 = my_app(v)
```

operators on AppFutures:

```
d = app_that_returns_a_dict()

future = another_app(d['whatever']) # breaks (before)
```

Story 6: Non-language Features

High Throughput Executor

- drain time

- explicit cpu affinity mode (when the cpu affinity patterns are complex) [*christine]
- multiple workers sharing one CUDA device [*christine]
- MPI mode (MPIExecutor) [*yadu]

providers:

kubernetes: more configurability
slurm: use accounting database for reduced load

files:

zip file staging
stdout/stderr staging
configurable
auto-naming

RADICAL-Pilot
executor

richer usage collection
[*Nishchay]

Story 7: Contributing groups

Globus Compute
Cooperative Computing Lab at Notre Dame - Work Queue and Task Vine [*Doug, Colin]
LSST Dark Energy Science Collaboration - funds 25% of my time, monitoring/reliability
Outreachy - story 2
NERSC - SLURM
ALCF [* Christine]
Debian Project - packaging maintenance
Rosen Research Group at Princeton [* Andrew]
Center for Molecular Modeling, Ghent University [* Sander] -
RADICAL research at Rutgers - new executor
Flux Framework Community - existing executor
UCL Advanced Research Computing
Woodwell Climate Research Center - kubernetes
National Snow and Ice Data Center

Don't forget the non-PR contributions! [* Dan]