

The Parsl FluxExecutor

James Corbett

October 28, 2021



The FluxExecutor brings Flux's scheduling to Parsl

- Flux is a next-generation resource manager and first-class scheduler
 - Developed at LLNL
 - Plan of record is that Flux will be the system resource manager on El Capitan, LLNL's 2023 exascale machine
 - Flux is unique in that it can run *within* another resource manager, like Slurm, Cobalt, or LSF
- The FluxExecutor is a Parsl Executor class that interfaces with Flux
 - Parsl apps are submitted as Flux jobs
 - Flux handles all of the scheduling and execution of the jobs and reports back to Parsl



Flux recently won a 2021 R&D 100 award

The FluxExecutor's intended uses

- The FluxExecutor specializes in applications that require a nontrivial resource set (like MPI or other compute-intensive applications), and collections of applications with highly variable resource requirements.
 - Flux's sophisticated scheduling will handle those applications logically and efficiently.
 - The scheduling policies are customizable.
 - By default, apps get exclusive access to the CPU and GPU resources they request.
- Flux provides a portable way to launch MPI applications
 - Except last-generation Cray MPI (Trinity/Theta)
 - Workaround is to build all applications (including **mpi4py**) with another MPI (e.g. OpenMPI)
- Each app passed to the executor can specify a number of options for requesting CPUs, GPUs, and MPI ranks, and also for tweaking their placement and distribution.

FluxExecutor future work

- Make Flux and the FluxExecutor widely available as part of the ExaWorks SDK
 - As it stands, users will need to build Flux themselves to use Flux and the FluxExecutor
 - Except at ORNL and LLNL, which have public Flux deployments
 - But Flux doesn't have many dependencies
- Get the FluxExecutor in a tagged Parsl release
 - it isn't in Parsl 1.1, the latest release
- Run some benchmarks on a large machine, like NERSC's Perlmutter

Links

- Flux home page: <http://flux-framework.org/>
- Flux documentation: <https://flux-framework.readthedocs.io/en/latest/quickstart.html>
- FluxExecutor interface documentation:
<https://parsl.readthedocs.io/en/latest/stubs/parsl.executors.FluxExecutor.html>
- FluxExecutor example configuration:
<https://parsl.readthedocs.io/en/latest/userguide/configuring.html#toss3-llnl>
- FluxExecutor blog post: <https://parsl-project.org/2021/06/30/Parsl-Flux.html>



Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.