Contribution ID: **569** Type: **presentation**

Interactive, scalable, reproducible data analysis with containers, Jupyter, and Parsl

Monday, 9 July 2018 12:00 (15 minutes)

In the traditional HEP analysis paradigm, code, documentation, and results are separate entities that require significant effort to keep synchronized, which hinders reproducibility. Jupyter notebooks allow these elements to be combined into a single, repeatable narrative. HEP analyses, however, commonly rely on complex software stacks and the use of distributed computing resources, requirements that have been barriers to notebook adoption. In this presentation we describe how Jupyter can be combined with Parsl (Parallel Scripting Library) and containers to enable intuitive and interactive high performance computing in Python.

Parsl is a pure Python library for orchestrating the concurrent execution of multiple tasks. Parsl is remarkable for its simplicity. Its primary construct is an "app"decorator, which the programmer uses to indicate that certain functions (either pure Python or wrappers around shell programs) are to be treated as "apps." App function calls then result in the creation of a new "task"that runs concurrently with the main program and other tasks, subject to dataflow constraints defined by the availability of app function input data. Data dependencies can be in-memory objects, or external files. App decorators can further specify which computation resources to use and the required software environment to run the decorated function. Parsl abstracts hardware details, allowing a single script to be executed efficiently on one or more laptops, clusters, clouds, and/or supercomputers. To manage complex execution environments on various resources and also to improve reproducibility, Parsl can use containers—lightweight, virtualized constructs for packaging software with its environment—to wrap tasks.

In this presentation we 1) show how a real-world complete HEP analysis workflow can be developed with Parsl and 2) demonstrate efficient and reproducible execution of such workflows on heterogeneous resources, including leadership-class computing facilities, using containers to wrap analysis code, Parsl to orchestrate the execution of these containers, and Jupyter as the interface for writing and executing the Parsl script.

Primary authors: Mr BABUJI, Yadu (Computation Institute, University of Chicago and Argonne National Laboratory); Dr CHARD, Kyle (Computation Institute, University of Chicago and Argonne National Laboratory); Dr FOSTER, Ian (Computation Institute, University of Chicago and Argonne National Laboratory); Dr KATZ, Daniel S. (ional Center for Supercomputing Applications, University of Illinois Urbana-Champaign); Dr WILDE, Michael (Computation Institute, University of Chicago and Argonne National Laboratory); Ms WOODARD, Anna Elizabeth (Computation Institute, University of Chicago); Dr WOZNIAK, Justin M. (Computation Institute, University of Chicago and Argonne National Laboratory)

Presenter: Ms WOODARD, Anna Elizabeth (Computation Institute, University of Chicago)

Session Classification: T6 - Machine learning and physics analysis

Track Classification: Track 6 –Machine learning and physics analysis