

Demo: Extending Jupyter to Support Interactive High Performance Computing

Matthew Henderson; Shreyas Cholia*; Oliver Evans; Fernando Pérez
Lawrence Berkeley National Laboratory

*Corresponding author address: 1 Cyclotron Road MS 943-2239, Lawrence Berkeley National Laboratory, Berkeley, CA, 94609, USA; email: scholia@lbl.gov

Abstract: *This demonstration will showcase our work in integrating the Jupyter platform with HPC resources, including extensions and modifications that enable “human in the loop” interactive supercomputing.*

1. Introduction

High Performance Computing (HPC) systems and workflows process and analyze data produced by large-scale experiments and simulations, such as first-principles materials structure calculations, supernovae simulations, and mass spectrometry image analysis. These largely focus on a non-interactive, asynchronous, batch execution process that can use thousands of cores and run for hours to days. Historically, these systems have not been designed to maximize the human utility and ease of interactive use, but rather to optimize for raw performance.

Simplifying and accelerating the mode of experimentation, which aligns with how scientists think and operate, is key to enhancing their productivity. This includes easy job submission and resubmission, introspection of jobs and their contents as they run, and easy ways of intercepting and manipulating data inputs and outputs for analysis and chaining of operations into pipelines. Introducing interactivity to scientific HPC applications and workflows provides a key missing human-in-the-loop capability to inspect the state of an execution in real time.

The Jupyter architecture (kernels, Notebooks, widgets, etc.) provides a solid foundation to address these challenges, and is already familiar to many scientists, but is missing certain key ingredients. Our work has been centered around extending the Jupyter platform to facilitate an interactive HPC experience for scientists by providing coupled widget-based and programmatic interfaces for interacting with batch jobs from the notebook. Our vision is to provide an interactive

Jupyter based-system that makes working on your laptop and working on a supercomputer a seamless experience that yields the best of both worlds, effectively “bringing supercomputing to your laptop.”

Jupyter notebooks provide visual communication of complex ideas and computations and can be used in place of or in addition to traditional script and code based scientific analyses and tasks. Formal workflows provide a structured mechanism to define a set of tasks and their control flow, and bring added benefits such as collection of provenance information that enhances their reproducibility.

We discuss how combining Jupyter notebooks with scientific workflows has the potential to dramatically increase reproducibility and collaboration among scientific researchers by allowing complex sequences of tasks to be communicated in an understandable and reproducible manner. Furthermore, we seek to extend the concept of the computational workflow to include interactive phases in addition to traditional non-interactive procedures. We discuss the implications of this type of human-in-the loop supercomputing on productivity and exploration in a scientific context. We will illustrate the power of coupling Jupyter with HPC systems and discuss how our system addresses some of the problems scientists face today, including:

- Managing notebooks with both synchronous local operations and asynchronous tasks that run on the supercomputing cluster
- Interaction with workflow managers, batch systems and job schedulers
- Dashboard interfaces for live plots and text output

We have designed a new component responsible for facilitating job control and communication in Jupyter. For communicating

with backend processes, we will provide a framework that can link to code and handle runtime data fetches.

We plan to demonstrate the proof of concept we are developing towards this end (part of the LBL LDRD project titled “Scaling Interactive Science for Data-Intensive Discovery”). We will

focus on specific use cases from Materials Science and describe our current progress and future plans for integrating interactivity and HPC systems in Jupyter.

2. Figures

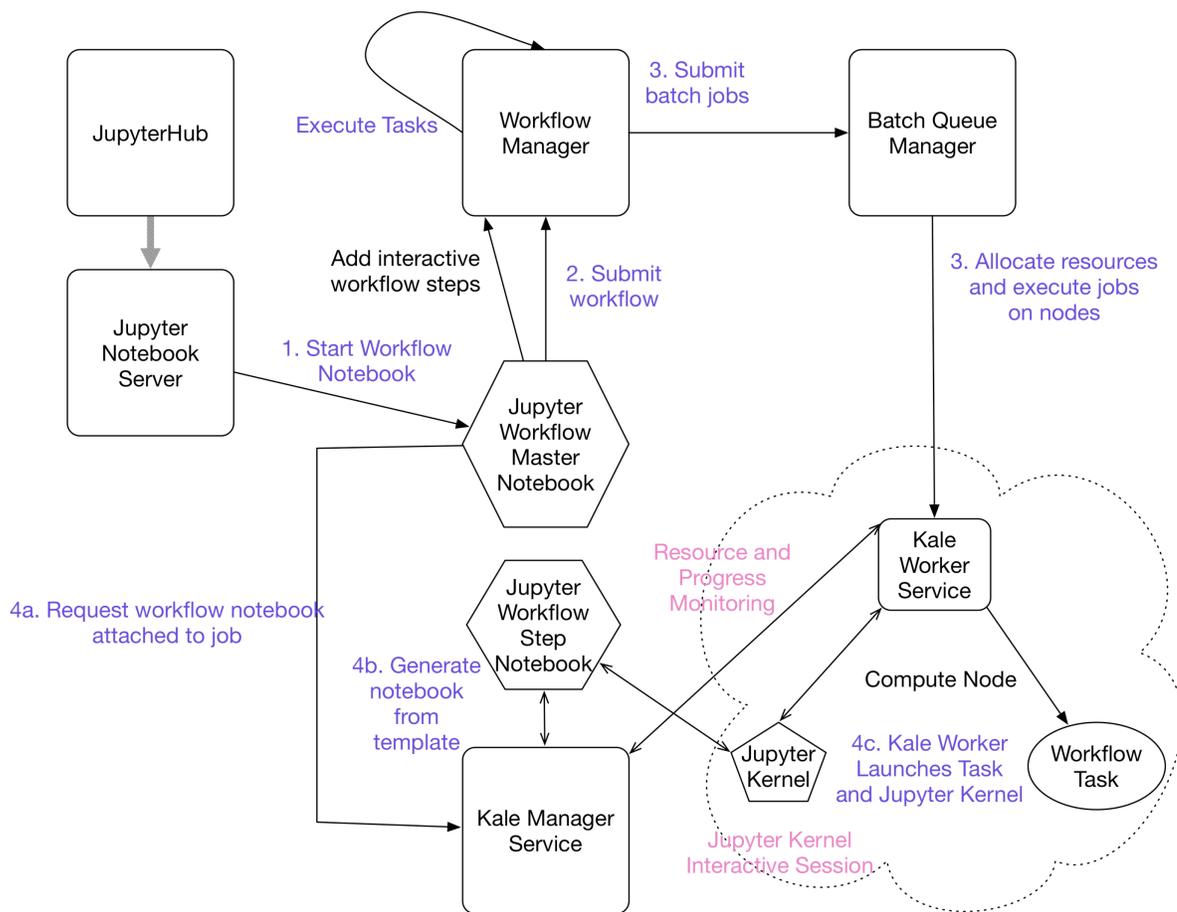


Fig. 1. Demonstrates the underlying architecture of our system.