An Extensible Ecosystem of Tools Providing User Friendly HPC Access and Supporting Jupyter Notebooks

Ben Glick and Jens Mache
Lewis & Clark College, Portland, OR
{glick, jmache}@lclark.edu

Introduction and Motivation

High Performance Computing (HPC) is becoming more and more important in many domains. However, HPC can be complex for people from domains other than Computer Science.

Our response: Create a “one stop shop” so that users only need to interact with one resource for all needs.

Abstract

- High performance computing systems can be hard to use, and can require advanced knowledge of the command line, and resource managers.
- In this project, we describe our extensible ecosystem of tools which has allowed students and researchers with minimal HPC backgrounds to use our system with virtually no training.
- Among the tools we support are
  - (1) a web-based job submission and management system,
  - (2) a GUI-based file transfer and sharing tool, and
  - (3) a Jupyter notebook environment.
- Together, these tools and the interfaces around them allow users to run complete HPC workflows without any use of the command line. Performance measurements show that our ecosystem incurs marginal overhead.

Architecture and Design

- The three component modules are the execution manager, the filesystem manager, and the external data manager.
- The interface ensures that all tools which are to become modules in our system are able to interact with each one of these components.
- Before a tool is added to the system, a "wrapper" layer needs to be written for it.
- This allows the tool to interact with the rest of the ecosystem and ensures that it follows the interfaces provided by the ecosystem.

Usability and Evaluation

- Our ecosystem has already been used for multiple computational science projects.
- Our users, both professional researchers and students, have found our ecosystem to be easier to use than they expected.
- Users have a simple, intuitive workflow which allows them to design, implement, execute, analyze, and share their HPC tasks easily.

The system has a web-based GUI form as well as a RESTful interface for automated job submission.

Our Jupyter notebook installation is simple and easy to use. As an example, this is a code sample which calculates pi via Monte Carlo simulation through our Jupyter notebook.

Summary of benchmarking results showing that running HPC jobs through our ecosystem only imparts a slight performance penalty, done using repeated matrix multiplication submitted to our cluster both traditionally and through our ecosystem’s Jupyter Notebook server.