Developing Interactive Jupyter Notebooks for SDSC HPC Systems

Dhruv Kumar, Leo Gu, Sarah Xie, Mary Thomas

August 18, 2020

Abstract

The goal of this research project is to develop a set of Jupyter Notebooks that can be used to train users on the Comet supercomputer. These notebooks will be hosted on the San Diego Supercomputer Center (SDSC) GitHub.io training pages. The purpose of this project is to understand the basics of High Performance Computing, Jupyter Notebooks and GitHub. For this, we worked with existing projects to develop new notebooks. The research components are to contribute to the body of knowledge needed for hosting live, dynamic, interactive services that interface to HPC systems, and to learn how to develop interactive notebooks to be used for education and training of the users of these systems. As a result of our efforts, we produced several new Jupyter Notebooks that will be used for Comet/SDSC training.

1 Introduction

One of the inherent issues with High Performance Computing is that the learning curve for scientists looking to actually apply it to their research is too high. There are usually too many levels of understanding between the scientist who does not want to sink their time into learning about how to use a supercomputer and the actual usage. That's why we resorted to a Jupyter Notebook interface which is a lot easier in terms of basic coding and should allow the scientists to still run all of their models on the supercomputer. Jupyter Notebooks feature a nice clear interface, but also allow you to use all the basic modules that you can download in python and the ability to run the bash shell and more. Thus, the idea of a shared Jupyter Notebook platform was created allowing multiple people to be able to be on the same Jupyter Notebook at the same time. The one issue with this is that the connection to JupyterHub is HTTP and not HTTPS which means that it is not secure because there is no encryption. To go around this, the reverse proxy service allows for a secure connection to this Jupyter Notebook.

2 Methods

In order to create a persistent JupyterHub web service, we learned about batch scripts and launching Jupyter Notebooks through Conda and Linux. Using the interactive software that Jupyter supports, we designed a series of tutorials that guide users through Python basics, running jobs using the Comet Slurm manager, and real-world applications using Python. We then studied High Performance Computing (HPC) and used our experience of the bash environment to launch the Jupyter Notebooks on both laptops and Comet. Subsequently, we did testing for the reverse proxy service which allows users to securely launch a Jupyter Notebook on Comet. Lastly, we researched different applications of Python and individually created Jupyter Notebooks on various packages and applications that branched into different fields of science such as machine learning and bioinformatics. As part of our learning curve, we learned to use the Atom IDE, and GitHub repositories to store our work and to clone onto different machines.

2.1 Python Packages

To show package support and potential uses, we created two notebooks using packages that have long-reaching potential and uses in the field. Part of these packages is an introduction to the basics with the basic documentation, but to go even further, the notebooks provide interesting applications of Networkx and Pillow. Networkx is a network creation module that can be used for anything from bioinformatics to airline traffic. The module has an extremely

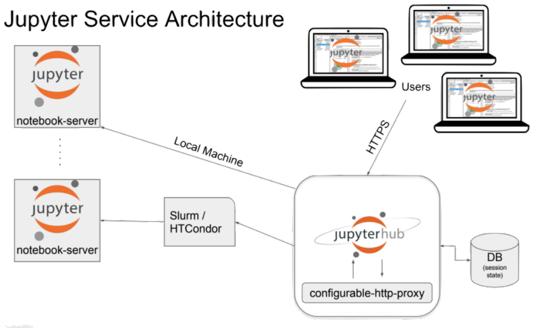

Figure 1: Jupyter Service Architecture

important use on a supercomputer level too because of the fact that a lot of the data that is going to be stored is going to be massive, computation, and storage wise. Pillow on the other hand demonstrates image processing, allowing you to manipulate image files to make data clearer and more readily available to access. The potential for scientists who are interesting in applying these tools into Comet allows for an engaging and more interactive experience with the Jupyter Notebooks.

2.2 Real-World Application

In the process of creating test notebooks for learners to utilize after connecting to Comet, we wanted to provide notebooks that could be interesting to a variety of individuals. Part of this included demonstrating the computational power in which Jupyter Notebooks could be utilized to compute data for the sciences.

For this purpose, a folder of environmental-science based notebooks was selected. The notebooks provide computable climate data and analysis with the purpose of allowing the user to recalculate and reproduce results and analyses using raw data. This is all outlined in the folder's README. The notebooks themselves draw data from multiple sources. Using the data, the notebooks create graphs, calculate results, and explain the implications of the data. Through this lengthy process of analyzing the raw data, conclusions about the climate can be drawn. These notebooks are adapted from Certik on Github for the purpose of providing test material for those connecting to Comet that is interesting to those involved with environmental science.

Another notebook project that was worked on included the applications of Python on real-world forums. One such forum was the Bank of New Zealand Forum, where users discussed questions regarding banking and credit cards. The notebook aimed to teach users about JSON files, web scrapping, and creating a simple recommendation model. Users extract the JSON files from the forum by taking advantage of the infinite scroll feature in many Discord forums. With this feature, the posts are simply on different pages, with each page containing the JSON script for the posts on that page. Thus, the python code is able to go to each page and pull the JSON script. Next, information about each post was extracted using the Beautiful Soup Python library and parsing through the JSON files. Finally, a recommendation model was built by creating a pairwise matrix containing the cosine similarity of all the questions. Thus, the lower the score, the more similar the questions in the post. The top 5 posts with the lowest scores were given as posts that are similar to the current post.

2.3 Python technique examples

To create engaging notebooks that are able to provide dual value to its readers, two of the notebooks that we created demonstrated the ways in which Python can be used to solve problems using interesting techniques.

One of these techniques is a programming strategy that dates back to the 19th century when Dedekind utilized the concept of recursive functions to define and analyze the natural number. This age-old strategy can now be used in Jupyter Notebooks using Python. The process of recursion involves a function calling itself in its execution. The notebook teaches the reader the basic concepts of recursion, including its need for a recursive case and base case. Using the example of factorials, in which the function calls itself with an incrementally smaller parameter, the program uses recursion to solve factorials (which can be defined recursively in mathematics). To further extend the reader's knowledge of recursion, an example in which recursion is used to solve a maze is used. The program recursively checks for routes, thus finding a way in which it can solve a maze. This notebook was created with the purpose of teaching those who connect to comet recursion and some of its applications.

With large sets of data in today's research, an increasingly important skill is to be able to cluster data and extract representative models of the entire data set. Utilizing python packages from scipy and sklearn, K-means and Agglomerative Hierarchi-

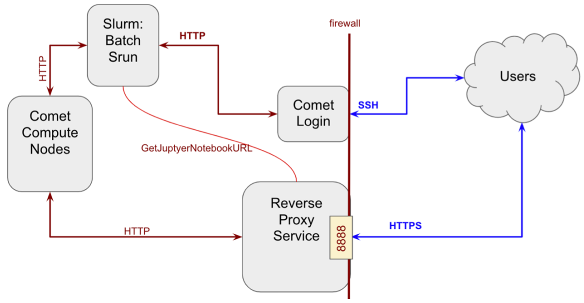

Figure 2: Reverse Proxy Service Architecture

cal Clustering were performed. Initially, random data sets were generated in a two-dimensional space for simplicity. Then, clustering was performed to extract 4 cluster centroids. This process of clustering aims to group similar data points together. An advantage of clustering a data set is that further analysis can be performed on a smaller data set using only the cluster centroids. When clustering is done accurately, centroids can be representative of the data set. By creating a notebook with a simple example, users able to copy and paste code for their own specific applications of clustering.

2.4 Jupyter Notebook Security

Currently, on the Comet HPC system, Jupyter Notebooks can be launched in an insecure fashion. That is, they will open in the browser with HTTP and are prone to hackers and invasion of privacy. A more secure method of launching Jupyter services is Reverse Proxy. When users run the notebook for the reverse proxy, they are assigned a random compute node, through which their connection is secure. Thus, when they open the Jupyter Notebook in a web browser, they will see HTTPS, where S stands for secure. We aimed to improve the use of the reverse proxy and tested the service numerous times to ensure that it was working properly. We provided the mentors with feedback regarding the clarity of the instructions so that more users would open to using the secure method of launching Jupyter Notebooks.

3 Results and Discussion

Through our internship, we learned skills specifically in logging in through SSH, creating Jupyter Notebooks, and launching Jupyter services through the Comet HPC system. We have become more aware of the educational challenges that face scientists in the research community. In our process of creating the Jupyter Notebooks, we noticed a few tips that enhance the readability and usability of the notebook for users. 1) A list of Python packages that need to be installed should be listed at the top of the Jupyter Notebook. While this might seem trivial, packages that require certain dependencies might not be so obvious to a novice. 2) Python code should be ready to run or be copied. When the notebook is meant to teach a user, it can be helpful if the user simply has to run the code for it to work. Sometimes, the user can learn easily when they see the output first. 3) Using Markdown to write comments about what the code is doing and why. This helps the user gain a deeper understanding of the code and allows them to reproduce the code for their specific case. We believe that these strategies should be implemented by those making Jupyter Notebook tutorials so that their clarity can be improved.

4 Conclusion

We started the REHS program with knowledge in Linux and operating systems but had minimal experience with HPC systems. However, our mentor, Dr. Mary Thomas, gave us guidance on how to navigate our way on the Comet HPC system. We went through many hours of tutorials and webinars to learn new skills and techniques regarding running jobs and Miniconda. At SDSC, we worked as a team to expand our understanding of High Performance Computing and the implementation of Jupyter Notebooks using Python. We also faced several challenges that we were able to overcome, including installing Miniconda, python package installation, launching notebooks on Comet, and spawning notebooks via a remote connection to Comet. Overall, we have truly enjoyed our experience in REHS and are fascinated by the complexity and brilliance of HPC systems and gave gained an appreciation for those who work hard to maintain and improve them. Although our time in the REHS program was limited, some potential future work includes expanding our tutorial library and developing a JupyterHub authenticator that is compatible with UCSD accounts.