Link the Future And Past User Data Analysis with Jupyter and Xcache

A Letter of Interest for Snowmass 2021 Computing Frontier

Doug Benjamin (Argonne National Laboratory, dbenjamin@anl.gov)
William Strecker-Kellogg (Brookhaven National Laboratory, willsk@bnl.gov)
Wei Yang (SLAC National Accelerator Laboratory, yangw@slac.stanford.edu)
Shuwei Ye (Brookhaven National Laboratory, yesw@bnl.gov)

BNL and SLAC each operate a user analysis facility for the US physicists of the ATLAS experiment. There we see a slow increase of user activities using today's Python based industry data science and Machine Learning tools. Some users also prefer to use Jupyter for interactive analysis instead of the traditional X-windows based analysis. On the other hand, ATLAS experiment's data products are in ROOT format, which is optimized for storage. ATLAS data files are also distributed around the world. At BNL and SLAC we are working on narrowing the gap between the decades old distributed data model and the power of Python and GPUs by investigating various software tools and environments consisting of Jupyter, PyROOT/uproot, GPU and ML resources, and Xcache to enable users to quickly go through a large set of official ATLAS data and then focus on final stage data analysis using modern methods.

We believe that the above issues aren't limited to the ATLAS experiments, and we are working on addressing them in a way that others in the HEP community can benefit (though the text below uses ATLAS as an example). Our current and future works comprises three main areas: (1) provide quick and easy access to data; (2) replace X-windows remote data analysis by Jupyter; (3) provide modern data science and machine learning tools in Jupyter, backed by GPU resources. While we are working in these areas, we will make sure this JupyterLab/Xcache environment is able to accommodate some of the recent R&D works by others on newer, non-ROOT data formats, and will make the environment portable to HPCs, clouds, or possible super analysis facilities.

Xcache enables us to use a small, high performance storage to accelerate data access. The integration of Xcache with the ATLAS data management (DM) system Rucio frees users from having to know where the input data is (It is also possible to integrate with DM systems from other experiments). It also eliminates the need to follow changes of the data location. By providing a list of location-invariant logical file names to the Xcache, users can always assume that Xcache has the data or can find the data. Because Xcache only fetches blocks of a data file upon request, users can quickly skim through a large amount of data with only a fraction of the total data volume being passed over the wire.

We are aware of the R&D activities to bring HEP data to users in non-ROOT, possibility column-based format. Our current strategy is to either run services at BNL and SLAC when these R&D produce relatively mature software products, or if those services run outside of BNL and SLAC, we will investigate accessing those new data formats via Xcache.

Increasingly the interactive data analysis has to be done at a remote location such as analysis facilities due to the amount of resources required. Traditionally ROOT-based interactive data analysis is done over remote X-windows display. Jupyter can replace the X-windows based analysis, with very little latency or network-induced lag, and yet provide notebook capability. At BNL and SLAC, several Python 2 kernels in JupyterLab are integrated with ATLAS analysis software releases, and support uproot, PyROOT and ROOT C++. The latter two can also read ATLAS xAODs data files. This environment offers a smooth transition from X-windows based ROOT data analysis to Jupyter based ROOT data analysis. Future work is needed when newer ATLAS releases transition to use Python 3, and when ATLAS moves to use the PHYS or PHYS_Lite data format.

The JupyterLab at BNL and SLAC also includes Python 3 kernels that support CUDA based GPUs and ML packages such as Tensorflow and Keras. Along with uproot and PyROOT, this is a future-oriented environment where we will closely follow changing user-demands. While it is impossible for us to provide all the Python 3 packages users need, we focus on providing key packages that require close integration with our hardware environments, as well as key packages that enable users to add additional packages by themselves.

The JupyterLab environment at BNL and SLAC can also scale up horizontally. This is currently being implemented in two ways: (1) Terminal windows in JupyterLab allows users to directly submit batch jobs from the command line; (2) DASK integration with batch systems allows Python to spread the calculation beyond the currently allocated Jupyter resources via batch jobs. In the future, through federated identity, it is possible that DASK (or similar tools) will allow these interactive works to burst to the resources at remote HPCs or other facilities.

One design principle of this JupyterLab/Xcache environment is to contain a relatively complete runtime base environment, in order to be independent from the hosting site's environment (except hardware). Already we can run all components in non-privileged containers or run in virtual environments. Our goal is also to be able to run the JupyterLab/Xcache environment from small scale to large scale, from desktops to HPCs, and at a super analysis facility when it becomes a reality.