Toward Interactive Supercomputing at NERSC with Jupyter

Rollin Thomas, Shane Canon, Shreyas Cholia, Lisa Gerhardt, and Evan Racah

May 9 2017
Data Science Process

1. Reality
2. Raw Data Collected
3. Data Is Processed
4. Clean Dataset
5. Exploratory Data Analysis
6. Models & Algorithms
7. Data Product
8. Communicate Visualize Report
9. Make Decisions

Diagram courtesy of “Farcaster” at English Wikipedia
Data Science [Wikipedia Definition]

- Get manageable chunk of data and copy it to your laptop/workstation
- Write code/scripts, make diagnostic plots, construct and test models
- Loop is very short between thinking up a query and executing it on data
  - Real-time testing of models that explain the data
  - Real-time feedback in the form of plots and results
  - ... hard to keep it all organized and explain what you did
Enter IPython and Jupyter

- **IPython**: Side project that grew into a data analytics phenomenon.
- **IPython Notebooks**: *Literate Computing, “Narratives”*
  - Code and comments: Reproducibility, show your work!
  - *But wait there’s more*: Rich text, plots, equations, widgets, etc.
- **Jupyter**: Language agnostic “notebook” part of IPython

Diagram courtesy of “Farcaster” at English Wikipedia
Why Jupyter@NERSC?

NERSC is the production HPC & Data Facility for Department of Energy Office of Science

- Largest Federal sponsor of basic research in the physical sciences.
- Lead Federal agency supporting fundamental scientific research for our Nation’s energy future.

Bio Energy, Environment

Advanced Computing

Materials, Chemistry, Geophysics

High Energy Physics

Nuclear Sciences

Fusion, Plasma Physics
Cori: Friendly for “Data Users”

- Two architectures in one system:
  - **Data**: 2388 nodes, 32-core Intel Xeon “Haswell”, 128 GB DDR4
  - **HPC**: 9688 nodes, 68-core Intel Xeon Phi “KNL”, 96 GB DDR4 + 16 GB MCDRAM
- Haswell login and **special-purpose large memory nodes** (512 & 768 GB)
- NVRAM Burst Buffer for IO acceleration
- Shared and real-time queues
- Shifter for containerized HPC
Why Jupyter@NERSC?

Deep Questions  ➔  Expensive Detector Technologies  ➔  Insightful
Instruments/Facilities
High-bandwidth Networks
Simulations
Real time predictions?
Exploratory analysis?
Decision making?
Expose, Integrate NERSC Resources

**Submit, Monitor, Interact**
- Batch Queues
  - `sbatch`, `squeue`, `srun`, `sacct`

**Query, Analyze, Visualize**
- NERSC Global File System
  - `/project`, `$SCRATCH`, `$HOME`

**Standardize, Reproduce Results**
- Database Servers
  - `mongodb01...`, `scidb1...`
- Software Environment Modules
  - `python/2.7-anaconda`, `python/3.5-anaconda`
Central Role of Python at NERSC

Python is the most popular language at NERSC used to:

- Script workflows for both data analysis and simulations
- Perform exploratory data analysis
Customizing Jupyter, Sane & Safe

- Users customize their notebooks with libraries and APIs of their own design or from third parties.
- NERSC wants to offer Jupyter to users so they don’t set it up themselves in an insecure way.

Example PyROOT Kernel Spec

```json
{
  "display_name": "HEP",
  "language": "python",
  "argv": [
    "/global/common/cori/software/python/2.7-anaconda/bin/python",
    "-m",
    "IPython.kernel",
    ",",
    "{connection_file}"
  ],
  "env": {
    "LD_LIBRARY_PATH": "/usr/common/software/root/6.06.06/lib/root",
    "PYTHONPATH": "/usr/common/software/root/6.06.06/lib/root"
  }
}
```
Jupyter@NERSC
Evolution of Architecture
First Architecture: “Edge Service”

August 2015:
- Single Docker container with access to NERSC Global File System
- Very popular service: 100+ users
- Missing:
  - Access to Cori Lustre Scratch
  - Interactivity with Cori batch queues
  - Cori Python environment.

Projects:
- OpenMSI
- Metabolite Atlas
- LUX
Second Architecture: Cori Login Node

August 2016:
- Standalone Hub server in **Docker**
- SSH spawner spins up notebook on special-purpose Cori login node
- Access to Cori Lustre Scratch
- Same Python environment as Cori login
- Interactivity with batch queues

Projects:
- LSST
- Metabolite Atlas
Our Extensions to JupyterHub

- **jupyterhub.auth.Authenticator**
  - Use MyProxy to login to NERSC CA server with user/pass to get X509 certificate credentials.
  - No need to run JupyterHub with additional privileges, or root access.

- **jupyterhub.spawner.Spawner**
  - SSH to Cori with user’s credential.
    - Uses GSISSH, but can use SSH.
  - Notebook starts up, spawner goes away, Notebook communicates w/Hub, keep PID.
SLURM MAGIC

- Jupyter “%magic” commands:
  - Expose extra-language functionality
  - Outputs are first-class Notebook objects

- Developed wrappers around SLURM commands. [https://github.com/NERSC/slurm-magic](https://github.com/NERSC/slurm-magic)

- `%squeue`
  - %squeue -u rthomas

- `%sbatch`
  - %sbatch script.sh

- `%%sbatch`
  - `%%sbatch -N 1 -p debug -t 30 -C haswell`  #!/bin/bash
  - `srun ...`
LIVE DEMO: What Could Go Wrong!? 
In Development: Cori Computes

Cori Login Node

- Notebook Server Process
- Kernel Process

Cori Compute Node

- Notebook Server Process
- Kernel Process

Cori Compute Node

- Notebook Server Process

Cori Compute Node

- Kernel Process

Role of SDN after Authentication
The Ultimate Jupyter@NERSC

**Software defined networking**

*Advertise IP of notebook server back to user.*

*Notebook on login node, kernel on compute.*

*Notebook+kernel on login, Spark job on computes.*

**Leveraging interactive QOS**

*Immediate access to compute up to four hours.*

**Shifter**

*Customize notebook/kernel’s environment.*

*Make larger-scale analytics apps actually start up.*

**Other possibilities**

*Notebook/scheduler on Haswell, kernels on KNL?*
Customizations to Jupyter

https://github.com/jupyterhub/batchspawner
https://github.com/jupyterhub/wrapspawner

Customize Access
- Burst buffer for your job?
- Cori node or compute?

Customize NERSC UX
- “My Shifter images”
- “My favorite job templates”
- ...
Who is Responsible?

NERSC
- Data and Analytics Services Group
- Security and Networking Group
- Computational Systems Group
- Infrastructure Services Group

LBL Computational Research Division
- Usable Software Systems Group

Developer Community
- Jupyter Developers
- MSI, TACC, SDSC
Conclusion

- Jupyter is a powerful tool for exploratory data analysis that is increasingly popular with NERSC users.
- We anticipate that more users will be asking for tools like Jupyter, and for the data sets they analyze to be getting larger, requiring multi-node Jupyter jobs.
- We are working to find ways to scale Jupyter up to handle bigger data sets and interoperate with NERSC resources and environment.
- Thank you!
National Energy Research Scientific Computing Center