Networking and Information Technology
Research and Development

The government seeks individual input; attendees/participants may provide individual advice only.

Middleware and Grid Interagency Coordination (MAGIC) Meeting Minutes[1]
June 5, 2019, 12-2 pm
NCO, 490 L'Enfant Plaza, Ste. 8001
Washington, D.C. 20024

Participants (*In-Person Participants)

Ed Berger (Purdue)	Joyce Lee (NCO)*
Wes Bethel (LBL)	Brian Lin (UW-Madison)
Tom Brown	David Martin (ANL)
Richard Carlson (DOE/SC)	Valerio Pascucci (Utah)
Alan Chalker (OSC)	Gilberto Pastorello (LBL)
Shreyas Cholia (LBL)	Don Petravick (NCSA)
Kaushik De (UTA)	Steve Petruzza (Utah)
Florence Hudson (NE Big Data Innovation Hub)	Birali Runesah (UChicago)
David Hudak (OSC)	Alan Sill (TTU)
Brian Johanson (PSC)	Sonia Sachs (DOE/SC)
Margaret Johnson (NCSA)	Nathan Tallent (PNNL)

Proceedings
This meeting was chaired by Richard Carlson (DOE/SC). May 2019 meeting minutes were approved.

Speaker Series: Data Life Cycle
- Alan Chalker, Director of Strategic Programs, Ohio Supercomputer Center, *Open OnDemand Overview*
- Shreyas Cholia, Group Leader, Useable Software Systems, Lawrence Berkeley National Laboratory, *Jupyter – An Interactive Platform for Scientific Computing and Data Analysis*

Data Life Cycle Series Planning
Alan Chalker, David Hudak (co-PI0 - *Open OnDemand Overview)*
Platform developing with University of Buffalo and Virginia Tech; supported by NSF
Open, interactive HPC via the web; web based access to supercomputers
2 categories enabled by Open OnDemand:
- Interactive Applications: scientific applications that would like to engage with
- Cluster Access: traditional activities interact with cluster

[1] Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Networking and Information Technology Research and Development Program.

Architecture:
Running per user NGINX:

- Once logged in, everything being exposed to you is run as you in the system; i.e., whole stack. Easy to ensure access based on traditional Unix group membership and policies.
- Client can come in via any web browser; desktop is not required. Once connected to Apache front end that sets up reverse proxy; connects to special log in node on back end. Everything stood up on log in is run as individual user (includes NGINX).

Impact at OSC: (Diagram)
OnDemand brand (since 2012), although developed for 10 years. Lowers barrier to entry; faster for science

- OSC client community – OnDemand use increasing while non-on Demand clients decreasing
- Shorter delay in time for new client submitting first job when using OnDemand

Participating Institutions:
Nearly 200 downloads from RPM of Open OnDemand. Aware of 50 or so institutions (in production or testing for production) from federal labs, hospitals, educational institutions and businesses

Customizing OnDemand: Branding (Screen shots of landing pages; PSC, Bridges system)
Walkthrough: Everything on top bar can be customized (image)

- Files: File explorer – can transfer, edit, copy and paste on remote file system
- Cluster: in-browser SSH client will show up to give direct SSH (in-browser terminal emulater) access and system status
- Applications: Clients like to see queue and current load status of systems to submit job
- Jobs: 2 types – active job browser to show snapshots of job queue and job composer (form-based page allowing selection of templates)

Clusters: shell access to systems and system status (image)
Apps page: show type of apps (form asking different parameters)
Job composer: create new job from template; existing job or default template; can edit, submit -show job details.

Open OnDemand 2.0 Project Overview
4 main areas:

- Visibility: Open XCMod (XC metrics on demand); provides job accounting and metrics for systems; installed at many installations worldwide; tightly couple Open XDMod and XD Metrics on Demand (see graphs); show performance of nodes on job
- Scalability: support more types of computing resources
- Accessibility: appeal to more scientists from different domains
- Engagement: building community (e.g., MAGIC), making it a community-driven project

Staying in Touch

- http://openondemand.org; Discourse.
- If interested in evaluating Open OnDemand, will provide account
- Installation: can do 80% of installation remotely as nonprivileged user

Shreyas Cholia - ***Jupyter – An Interactive Platform for Scientific Computing and Data Analysis***

Jupyter:

- Guiding principles stem from concept of open, reproducible science
- Put ideas in notebook to be shared and communicated with other programmers and also incorporate in future work (able to annotate work); i.e., lab notebook
- Also ecosystem of people, tools and standards for interactive computing
- Scale up through iPython, which evolved into Jupyter (large team of OS contributors (industry, national labs, academia and global)).
- Definition: at core, driven by Jupyter notebook; tool for reproducible, shareable narratives, literate computing (enables annotate coding); explore data analytics, workflows through Jupyter
- Includes Rich Web Client – display text and math; enter mathematical notation in Jupyter; executable code and shareable
- How it works/Architecture: uses 0mq protocol (enables browser to talk to notebook server, which talks to language kernel (language interpreter); work stored in notebook file
- Core idea: based on HTTP (enables talking to servers) and web – Jupyter built on top; adds features
- Protocol: send any mime-type (e.g., images) and front end figures out how to deal with it
- Language-agnostic although originated with iPython, kernels for Julia, etc., that can interpret code and send back results using Jupyter protocol

Classic Notebook

- Jupyter notebook and file browser, terminal, and editor. Obtains basic functionality in context of web platform

JupyterLab

- Need new applications living alongside notebook. JupyterLab: common platform to host backends and run under Jupyter; can render different formats (e.g., Lorenz Differential Equations)
- JupyterLab Demo (has repos). Click on Launch Binder – grabs repo, packages and launch on container on Google cloud.
- Also Lorenz Differential Equations demo: allows live interactive visualizations; can plot results, etc. C++ kernel.
- Combining data with notebook (e.g., visualization of Zenome browser)
- Going beyond notebooks to bring in other applications (e.g., NERSC working on application to manage job cues)

Jupyter Hub: provides multiuser support for Jupyter. Set up to use own authentication.

- Centralized deployment: decide what users need to have in their system and how they want to run. Can spawn system in many different ways, including container-friendly (Binder uses K and docker; NERSC uses SSH)
- Can provide access to big data sets and can spawn notebooks on a post or platform that has direct access to datasets.
- Provide authentication that allows access to back end coding environment; ultimately, results in shared workflow and results.
- NERSC version allows launching of Jupyter at NERSC – Cori (workhorse); data and analytic services group (ML tools, optimizing on NERSC systems). Facilitating interactive computing at NERSC.

- Architecture:
    - Log into NERSC, which allows you to bring up node in Docker container (Spin- Docker platform for launching edge services- gives access to Jupyter notebook);
    - Can run Jupyter on reserved Jupyter nodes and their shared infrastructure; and
    - Can get own dedicated compute node and launch Jupyter there.
- SSH authenticator API to get tokens to do things; SDN also utilized
- Use Jupyter to manage parallel jobs:
    - launch notebook that manages parallel jobs
- Interactive distributed deep learning: hyperparameter optimization problem (film);
    - Jupyter notebook sets up parameters and training tasks, which runs in the background.
    - Pulling results from compute nodes in real time; can sift through results in real time (interactive: can start/stop selected jobs, tweak parameters and relaunch).
    - Great feedback from users

Reproducible Research (Binder)

- Provides complete software environment that can package up; put repo on GitHub, use tools to launch repo in cloud (through Docker, Kubernetes)
- Usage: 350k users in any given month (Oct 2018) – may be higher now

BinderHub: combination Jupyter hub +Binder

- Provides complete software environment – can package up; put repo in GitHub; and launch repo in cloud through Docker and Kubernetes
- Uses repo to docker tool- takes requirements file and generates own Docker file to come up with fully reproducible environment that you can capture; [builds and pushes image. Launch it into cloud].
- Open technology built on top of Kubernetes, cloud agnostic, scalable. (e.g., LIGO project: published gravitational wave event as set of Jupyter notebooks). Through Microsoft Azure Cloud, made analyses available to anyone who wanted to run it.

Jupyter and Education: interactive platform lends itself to training, presentation, live coding classes.

- UC Berkeley data science courses taught through Jupyter notebooks; students can run and follow along (see link). Datahub is backend tool for this. Good way to scale class.
- Wide industrial adoption of Jupyter. As large OS collaboration, funded through many sources (federal grants, industry)

LBL projects: Jupyter R&D

Jupyter designed to be modular and pluggable. More emphasis on integration of big data and compute. Looking for more input from community (e.g., reproducible science at scale, Binder HPC)

- How Jupyter can act as the primary super facility interface to drive experimental and observational data flows. Coming up with canonical notebooks for users to clone and re-run data flows and manage distributed workflows.
- Useable data abstraction projection: Using Jupyter and Juptyer lab to access remote distributed datasets
- Infrastructure: Building up robustness of Jupyter Hub and scalable to launch notebooks under different backend environments.

Jupyter Community Workshop: HPC and scientific facilities information exchange next week.

Discussion

- Fairly broad adoption across the world (e.g., CERN)
- Data integrity and security- Jupyter takes security seriously, talking to Thomas Mendoza (Lawrence Livermore) who is working on secure, end-to-end installation of Jupyter, from kernel to end user.
- Binder integration with Google – on paid basis. Many projects to integrate with HPC, Binder?
- Binder uses Kubernetes to spin up containerized environment. Roadblock: not have Kubernetes running on Cori system. Docker container environment uses Rancher, which is supposed to move up to Kubernetes; then can likely use Binder with Rancher. If have Kubernetes, can start looking into it.
- Assume Binder uses Kubernetes, but can't do it yet. Can implement Binder on own Kubernetes infrastructure. Look at BinderHub repo and Zero to Jupyter Hub repo are relevant; may just need to tweak a few things.
- Reach out to Shreyas with feedback. Can pass along specific use cases to Jupyter team. Jupyter can do a lot more in the HPC, large scientific workflows

July 3 Speakers

- Fran Berman, RPI, speaking on higher level organizational issues regarding stewardship and preservation;
- Invited: Victoria Stodden (reproducibility); Dan Katz (data publication; possibly Lorena Barber

Report

Dhruva Chakravorty (TAM) finishing containerization report. Hope to start DevOps report

Roundtable

Distributed collaborative, shared infrastructures not accepted for SC19.
Alternatively, propose BOF on the "Community Discussion on collaborative, distributed, shared and federated, scientific cyberinfrastructures". Loop panel (Dana Brunson(Internet2), Florence Hudson, (NE Big data innovation hub).

- Will circulate to Joyce Lee to send on to MAGIC.
- Florence Hudson added to panel.
- Rich Carlson contacting Arjun Shankar (ORNL); also contact Eric Lancon (BNL) – future lab computing working group discussing SC19 event.
- May lead to future workshop
- LSN overlap through GENI project (successful collaborative infrastructure)

PEARC19- July 28 – Aug 1

- AI for Good workshop; NSF cybersecurity for excellence workshop (Florence Hudson)
- Paper on ROI calculations for scientific and academic cyberinfrastructures (Craig Stewart)

Next Meeting: July 3 (12 noon ET)