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]
November 19, 2019, 1:30 MST
Supercomputing 2019
Colorado Convention Center
Denver, CO

Participants (*In-Person Participants)

Sadaf Alam (CSCS)*	Shantenu Jha (BNL)*
Yadu Babuji (ANL)*	Alexei Klimentov (BNL)*
Ray Bair (ANL)*	Eric Lancon (BNL)*
Debbie Bard (NERSC)*	Joyce Lee (NCO)*
Wes Bethel* (LBL)	Ketan Maheshwari (ORNL)*
Robert Bonneau (OSD)*	Daniel Murphy-Olsen (ANL/UChicago)*
Stuart Campbell (BNL)*	Lavanya Ramakrishnan (LBL)*
Shane Canon (LBL)*	Kamie Roberts (NCO)*
Richard Carlson (DOE/SC)*	Arjun Shankar (ORNL)*
Kyle Chad (ANL)*	Alan Sill (TTU/OGF)*
Vipin Chaudhary (NSF)	Cory Snavely (NERSC)*
Kuldeep Chawla (LBL)	Suhas Somnath (ORNL)
David Cowley (PNNL)*	Annua Surez (HPC Wire and Trivium Consulting)*
Eli Dart (ESnet)*	Nathan Tallent (PNNL)*
Zhihua Dong (BNL)*	Birali Runesha (UChicago)*
Kjiersten Fagnan (LBL)*	Tom Uram (ANL)*
Ian Foster (ANL/UChcicago)*	Rick Wagner (ANL)*
Richard Gerber (LBL)*	Sean Wilkinson (ORNL)

Proceedings

This meeting was chaired by Richard Carlson (DOE/SC) and Vipin Chaudhary (NSF).

CASC: Alan Sill – member organization of large scale computing organizations. Survey of those interested in advanced uses of data center and clouds throughout academic and research enterprise – open to all, especially those from historically, underrepresented institutions.

https://go.osu.edu/CASCsurvey

Deadline January 7, 2020. PEARC report and white pages will follow.

[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.

Arjun Shankar (ORNL), Eric Lancon (BNL), *Future Lab Computing Working Group (FLC-WG), Distributed Computing and Data Ecosystem (DCDE)*

Context: DOE/SC labs, both computational and experimental observational facilities, producing more data. Data rates growing. Demand on compute and analysis resources is increasing.

- Model today: big facilities looking for own sources of computing. Computational facilities (e.g.,NERSC) standing up resources for communities that use them.

- Connect some facilities together to take advantage of available cycles. Often there is available capacity despite oversubscription; opportunities to take advantage of computational resources across complex (e.g. Big Panda – used backfill cycles on Titan to advance science goals (almost doubled capacity in U.S.))

- Goal: Facilitate research without needing expertise into 1 team. Enable single PIs to go from beam lines and bring in data into compute environments wherever available in complex. Do analysis and use results in real time.

- 2017- Future Lab Computing working group (representation from labs, ESnet) observations in operations in each facility. Resulted in Report: Background and Roadmap for a Distributed Computing and Data Ecosystem and ensuing pilot.
- Pilot: implement key observation from report– Need bring in 5 major components to enable distributed computing and computing ecosystem. Building on lessons learned from grid; revisiting with modern technology:
  - Seamless user access – avoid difficult, custom processes of authentication
  - Coordinated resource allocations and cross-facility workflows – flow from end-to-end
  - Data Storage, movement and dissemination for distributed operations – easily move data
  - Variety, portability: through virtualization, and containers, etc.
  - Governance and Policy structures – governance across different facilities; resource allocation/prioritization.

Pilot:

Want to create more unified view into capabilities across complex. Test in focused pilot; multi-lab pilot team met for 1 year (ANL, BNL, LBL, ORNL, PNNL). Sought available, mature technologies to deploy

- Science driver
- Federated identity management – some inroads made for bringing users
- Portability across libraries – ran same container across BNL, ORNL and ANL – addressed some parts of portability
- Workflow through analytic notebooks – use Parsl implementation behind Jupyter notebook interface to facilitate workflows
- Data Transfer – relied on Globus

DCDE Overview = user from PNNL logged into id authorization set up in BNL and launch jobs locally through Jupyter interface and remotely through ORNL and ANL. Workflow across labs give scientists results on data collected.

Pilot Guidelines

- Simplicity (single research, small group)
- Lightweight, off the shelf components
- Easy, local installation
- Works locally and remotely, CL and web interface
- Portability
- Modern (sustainable) technology

Discussion

- Considered OSG-route, but identity mechanism and coupling with virtual organization, not ideal to allow some single PI experiments; not appropriate to connect experimental facilities to computational facilities. Discussed and debated.
- OSG can't explore all of technology space; room for innovative experiments. Time to go beyond single technology into multi-technology space.

Demonstration: David Cowley (PPNL)- role play as domain scientist

Proof of concept demonstration. Jupyter notebook used as interfaces.

Prototypical SC workflow – data coming off scientific instrument needing analysis to produce scientific results. Demonstrate pieces of workflow. Relion application – processes raw data to 3D structures of protein.

Use case – data coming off instrument. As single PI, seeking compute resources wherever can find. so

Will show multi-step, sequential workflow. Interacts directly with DCDE and using variety of compute resources.

Jupyter hub page->demo page.

Objectives – demonstration from single interface (JupyterHub and Python) – do computing and transfer data to and from multiple sites from single Jupyter notebook page. Parsl help deal with resource managers, storage resources and aspects of compute systems.

Using: compute resources of ANL's LCRC, BNL SDCC and DTN, ORNL CADES and DTN and Jupyter notebook page.

Software and infrastructure:

- InCommon, CILogon, CoManage - vary a bit from site to site, but all mapped to single DCDE Identity used.
- Globus & Oauth -use for identification
- Jupyter Notebooks: main interface
- Parsl: layer between notebook and compute resources. doing mapping and interface to high end compute interfaces

- Singularity container: same one used at each site to provide code/app running

Running batch job in each of these steps going into queue and return results per site

Discussion: Authorization occurs at each site; pre-authorized accounts.

Demonstration

1) Set up Parsl environment –specify how each of 3 compute sites is accessed.
2) Set up Globus and authorizations
3) Set of microscopy data – at BNL; push to sites to perform conducting
    a. 1st workload step – organizes raw data to prepares for Relion to work on it.
        i. Parsl executor in queue – waiting to get results.
        ii. Standard output indicates files organized and set up as needed.
        iii. Determining protein structures (freeze sample protein and put in electron microscope and run microscopy and take TB of data over multiple days on single experiment) – start characterizing and sorting them over workflow
4) Ran job at ORNL, pulled back to BNL and sync to ANL; sync back to BNL
5) Result- after extract job at ANL finishes, produces set of images (now can discern structure as application has started to identify individual particles and features in data set).
6) After more workflow steps, start to see images of protein viewed from different angles as many are in sample. Software tries to identify particle and orientation; further refines until start develop model of protein (18-20 steps in workflow). Large data set –takes days on nodes with dual groups
7) Final Result: list of output files from data set. At end, produce 3D model of protein in sample.

Want smaller resources that are available at DOE computing sites to be available to single PI.

Concerned about Parsl – wish to encapsulate Parsl in libraries

Summary:

Proof of concept for using multiple compute resources at multi sites from single Jupyter notebook under single identity.

- Used Parsl to provide interface between Jupyter and each compute resource used (could use other compute resources: could be local or from other compute sites). Parsl provides interface to use other compute resources
- Could bundle into set of libraries: API- that enables dispatch of this work to compute resources that want or need to target

Discussion:

- Ease of break live demo? Mostly works once push button; Parsl finicky
- Good start of showing why this matter and type of work can enable Distinction between this interface and batch type jobs (more computationally intensive) for OSG.
    - Simple use case chosen for this demo – to put together different pieces
    - 2 bulk use cases: 1) interactivity and exploring data 2) batch processing of data – both possible. Will work, but hasn't been demonstrated.

    - Want to submit jobs anywhere – hundreds of thousands of jobs – won't happen for Jupyter. Will the same technology stack be used for both (same authentication and reservation system); if so, need support of WBL and CWL and containers, so users can sync into it.
- Authentication token from web based authentication scheme and passing to underlying python libraries. Difficult to pass to CL – possible, but not as seamless.
- Interfaced to Slurm
- Valuable work. Obvious shortcoming in dealing with authentication and authorization goes back to early days. Worked on distributed method for authorization and account mapping. Not built for launching thousands of job.
- Need more inclusive process for evaluating other directions and on future lab computing group and explore other technologies and other efforts.
- Appropriate group for addressing needs of multi-lab computing?
- Addresses how group of users address code. Breaks a fundamental model of leadership facilities (2-factor code) – instead of fixating on technology but focus on use cases and how policies emerge – potentially building on it and let technology choices flow from it . Experiments critical in showing what current researcher can do in leveraging existing resources.
- Not large community effort – now trying to figure out how to expand and bring in more folks.
- NERSC – enthusiastic. Draw from diversity of parallel efforts, which are based on same principles (use existing tools, embrace community standards, automate workflows), avoid same mistakes. Aim forward.

<u>Europe</u>

- Phoenix project: Switzerland HPC center, etc. Started with Identity and access mgt. Identity management is easiest part. Issue arises when identifying service and what they can do and in what context. HPC systems lack infrastructure to implement cloud-based fine grained authorizations. Trying to solve some of these issues so can do some HPC and cloud type service authorizations.
    - Accounting and reporting: Who decides which resources get on which system?
    - Can bring in knowledge and experience in this context.
- EU Science cloud – working on identity access mgt and persistent ID (All about data - who can touch what data and when); focusing on these issues.
- Here, in demo, providing template on WF is useful.

LSN – embedded, distributed heterogeneous infrastructure. Manage control of system and integrate on demand. How to put in DevOps process. Would like to discuss further.

Resilience in infrastructure: recognizable and ready-made way to reconfigure system on demand. Automating the process is important.

JET group demo: look at infrastructure and network management. Suggestions to use techniques under cloud developed group and DevOps infra approach to dynamically make physical network infrastructure resilient in SDN, SDX type paradigm.

These issues are deeper than some things showing here. Hope some of work in creating service interfaces around resources in DOE complex will provide implementation ideas.

Future Lab Computing Working Group (FLC-WG)

DCDE Technical Elements

Overview- aims to unify and make available compute and data resources from DOE affiliated national labs to users. Talk focuses on technical components

Authentication, authorization issue

- InCommon Federation – chosen because of participation of DOE labs, but only provides SAML standard but no Oauth
- identify issues ->moved to CILogon (authentication hub for DCDE)
    - proxy linking to InCommon and provides X509 certificate service useful for integration with some services

Problem 2-

Registering users (CoManage service)- platform enabling user participation; web-based portal allowing user to identify self. Now being commercialized

Once 3 components set up, participating site roles:

Site admin:

AuthN/AuthZ: activities that each participating site must do (e.g., provisioning resources)

Creates appropriate mapfiles –map from Comanage to local user name in order to get on local site

Globus and oauth-ssh

Globus provides Oauth-ssh service over Oauth – users can ssh to sites via Oauth-ssh.

Discussion-

- If want to run 2 ssh services (either separate host or separate server) – not a Globus requirement. Standard ssh server with pan module.
- Globus currently uses Oauth tokens for almost all user authentication. Pan module receives Oauth token and authenticates user based on token. Since pan module fits into standard server open ssh server. A model to consider here is for-service or community accounts – could they be tied to pan module like this to allow authentication using Oauth, while other users come in through standard authentication vehicles?
- Can use SciTokens? Could package Oauth tokens in SciTokens but pan module currently wouldn't know what to do with it.
- How is token being passed? Client-side or through actual protocol.
- Opensource
- SSH -much potential. Not separate ssh server. Goal here is to enable many technologies, but when comes to APIs, ssh is potential API.

Globus transfer is used for data transfer purposes between participating sites

Technical requirements for Oauth-ssh

- Choose a port for each site to be opened
- Update DNS record at site so site itself gets authorized

Application and containers

- Chose Relion
- Containerized application using Singularity to unify troubleshooting, portability across sites; facilitated running same version of application across all sites.
- Each site runs same container and same version of Singularity

  Discussion – enabled encryption and digitalized

  Trusted application. Singularity- has mechanisms whereby portions of container can be digitally signed –>appealing when want to enable many run applications and want to trust applications executed

Jupyter and Parsl

- For interactive portion, chose Jupyter, which was integrated into DCDE project and DCDE users can log into. Only authenticated DCDE users can log in and perform work.
- Parsl- workflow platform, natural choice because python package that works well and well-integrated with Globus and Oauth

Summary

- Learning Curve: Challenge is many technical components. One approach: provide templated solutions to common user issues
- Site admin challenges (e.g., firewalls management very significant, installation and configuration of Oauth stack, JupyterHub etc.). Solution: Scripted several components

  Discussion

  - Firewalls are an issue even for default access from 1 lab to another, not just for DTNs. Orchestration and use of non-standard ports
  - Community perspective: can implement DTNs in SciDMZ; need similar effort for orchestration

Lessons Learned

ID Management

Federated ID mgt

- Need workaround at each site to accept users.
- CoManage: each member of DCDE has unique CoManage ID- can map to unique Unix account. Pre-provision accounts or create on fly (depends on site)
- Each CoManage groups may have various capabilities at each site. Need a way to publish these capabilities
- CoManage attributes may provide path to propagate user policy acceptance and training requirements

Portability across laboratories

- Chose singularity, but other technologies may arise. Compatibility of singularity versions at different labs.
- Simple users: may not know how to create container? May need service to encapsulate applications; submit automatically for users.
- Need to develop standard mode of communicating information

WF through analytic notebooks

- Parsl – simple, works locally, global, integrated with Jupyter. Still developing.
- JupyterHub- spend most time. Most difficult to solve (firewall exceptions). Works now…need mechanism to bridge systems to connect to other labs.

Data Transfer – not want to know (hidden)

- Note Globus – not commercial organization; not need license. Most users do not pay.

Information and policies

- Need standard procedures; and allocate resources to project
- Participants' expectations; Service level agreement?
- How to manage polices and priorities among groups and their members; what needs to be discussed

Final Discussion

- NERSC:
    - Agrees with lessons learned. Concern: managing policies and priorities between different sites (hardest area because not technical and no easy solution; requires many stakeholders and discussions)
    - Data movement – May more issues around moving data around and curating data – what is source of truth. Data at various sites – how managed; some issues may come up
    - These efforts reflect same fundamental questions. This type of project can expand in scope quickly and become paralyzed. Important to distinguish aspects of it. Important to make concrete, forward progress – break off manageable chunks (e.g., technical layer -- standards, interoperability issues)
    - Policy questions: Identify meaningful subset to start. Start small without overspecialize
- Data demos in SC14 – infrastructure items addressed: authentication, portable execution, etc. Still struggle. When started with ALS, struggled with how authenticate. Start with common way of crossing out authentication with DOE.
- Also look at HPC infrastructure design
- Action item: how to expand this effort to bring in right stakeholders going forward. Started with small pilot, but more to do. How to move forward in systematic way.

Next Meeting:

December 4 (12 noon ET): Please join MAGIC at its next public meeting to continue discussions.