ARISTOTLE
CLOUD FEDERATION

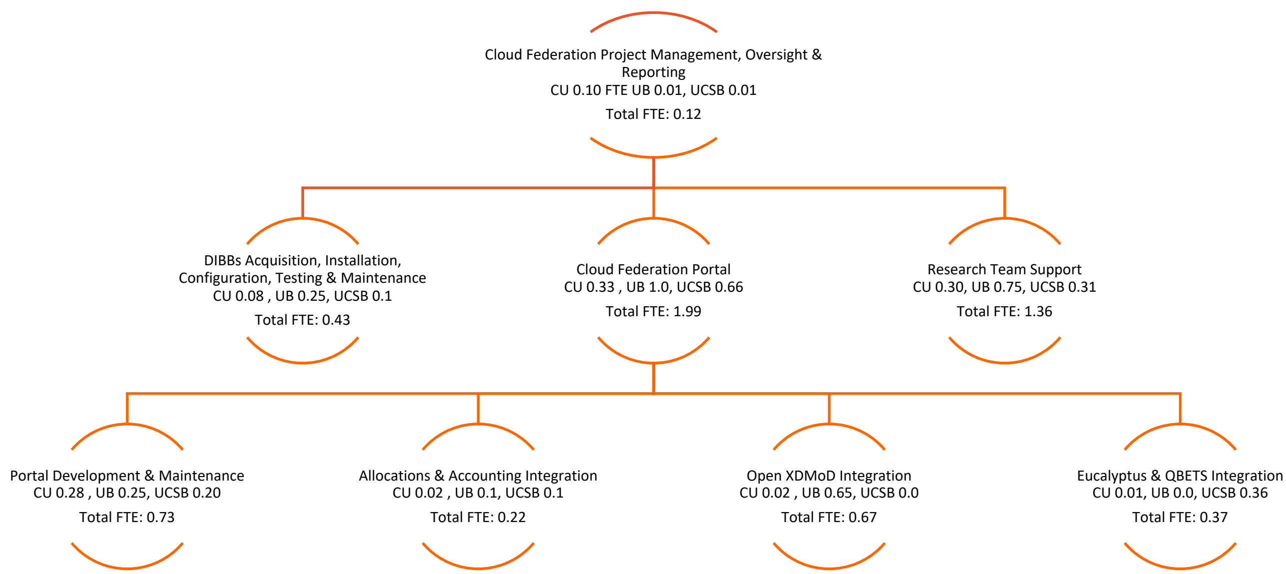
CC*DNI DIBBs: Data Analysis and Management Building Blocks for Multi-Campus Cyberinfrastructure through Cloud Federation

Program Year 4: Quarterly Report 4

9/19/2019

Submitted by David Lifka (PI)
lifka@cornell.edu

This is the Program Year 4: Quarterly Report 4 of the Aristotle Cloud Federation team. We report on plans and activities for each area of the project Work Breakdown Structure (WBS).



#1541215

Contents

1.0 Cloud Federation Project Management, Oversight & Reporting Report 3
1.1 Subcontracts 3
1.2 Project Change Request 3
1.3 Project Execution Plan 3
1.4 PI/Partner Meetings 3
1.5 Project Planning Meetings/Status Calls 3
2.0 DIBBs Acquisition, Installation, Configuration, Testing & Maintenance Report 6
2.1 Hardware Acquisition 6
2.2 Installation, Configuration, and Testing 6
2.3 Federated Identity Management 6
2.4 Cloud Status by Site 6
2.5 Tools 7
3.0 Cloud Federation Portal Report 7
3.1 Software Requirements & Portal Platform 9
3.2 Integrating Open XDMoD and DrAFTS into the Portal 9
3.3 Allocations & Accounting 10
4.0 Research Team Support 11
4.1 Science Use Case Team Updates 11
Use Case 1: A Cloud-Based Framework for Visualization & Analysis of Big Geospatial Data 11
Use Case 2: Global Market Efficiency Impact 12
Use Case 3: Application of the Weather Research and Forecasting (WRF) Model for Climate-Relevant Simulations on the Cloud 12
Use Case 4: Transient Detection in Radio Astronomy Search Data 13
Use Case 5: Water Resource Management Using OpenMORDM 14
Use Case 6: Mapping Transcriptome Data to Metabolic Models of Gut Microbiota 14
Use Case 7: Multi-Sourced Data Analytics to Improve Food Production & Security 15
5.0 Community Outreach and Education 15
5.1 Community Outreach 15
5.2 Education 15

1.0 Cloud Federation Project Management, Oversight & Reporting Report

1.1 Subcontracts
All subcontracts are in place. Nothing new to report.

1.2 Project Change Request
No new project change requests were made this quarter.

1.3 Project Execution Plan
The Project Execution Plan (PEP) was approved by NSF on 12/18/2015. We are operating as planned and continuously updating our PEP on a monthly basis.

1.4 PI/Partner Meetings
The results of partner meetings are included in Section 1.5.

- Met with Weston Maggio, RightScale manager, to explain the current goals of our open marketplace investigation and to further engage technical team support.
- Met with Sanjay Padhi, AWS head of research, and Stephen Fang, Google higher education. Secured public cloud credits for the project and discussed goals.

1.5 Project Planning Meetings/Status Calls

7/10/2019 status meeting:
- Held frequent meetings with RightScale to convey the goals of the cloud marketplace investigation and to better understand the capabilities of their Optima cloud cost management platform.
- Began experiments on costing the cloud. AWS services are running, but there are issues with paying with credits rather than dollars. We're working with Cornell's Cloudification team to resolve that. Also, we'd like to see hourly rather than end of day cost reporting. If RightScale isn't conducive to fine grain reporting and can only do specific tags, we may have to go down the road of parsing. We need to better understand how the cost allocation tags function and may have to look into custom tagging. We also need to dig into the billing API.
- Created instances with Google Cloud in Ansible, but the billing script is not terminating. Azure is reading activity out of our subscription but failing to authenticate into the RightScale API.
- Costing usage in multiple cloud platforms is challenging. We hope to eventually identify the problems, mask the pain, and write a white paper on how to make it less frustrating.

7/16/2019 status call:
- Began application kernel containerization work at UB. Application Kernel Remote Runner (AKRR) can use OpenStack as a resource and run Docker containers on it. We also have almost all application kernels in a Docker form (HPCC, HPCG, NAMD, NWChem, GAMESS) and ran them on OpenStack with Docker. Still need do some tweaking. For example, we used a NWChem container built by the developer that is slower than the version on our cluster; it was probably compiled without performance libraries. The question is: should we stick with the developers' version or cook up our own? We also have IOR and MDTest, but are fighting file system cache. While testing them with Singularity on our preproduction cluster, we got some unexpected results. In particular, sub-nodal jobs are faster with Singularity; we speculate that there might be an incorrect CPU affinity setting.

- UCSB is now using the Rancher software stack to support a new research group.
- Dartmouth is investigating Magnum, an OpenStack API service developed by their containers team to make Docker Swarm, Kubernetes, and Apache Mesos available.
- Cornell rejected a bogus request to be added to Aristotle 000 project.
- Use case support team and REU student are continuing with containerization development for public clouds. Telcon with Dartmouth to discuss their first use case: Jupyter notebooks for education to reduce campus computer labs which most of the time aren't being used. This is also of interest to the NSF NANOGrav project because they use Jupyter Notebooks to onboard researchers, from undergrads to senior scientists.

8/7/19 status meeting:

- Aristotle consultants got the Cornell REU student and Pat Reed's water resource management team up and running on Azure.
- We are increasing our skill set in AWS batch and parallel cluster. Met with Cornell Cloudification team (Marty Sullivan) to learn from their experience containerizing WRF for AWS and we're getting the transient detection in radio astronomy search data use case application running on AWS spot instances.
- Continuing to investigate issue with all public cloud credits not showing up in RightScale.

8/13/19 status call:

- Arnold Song, senior research software engineer, reported that Dartmouth has worked through most of their networking and hardware issues and is now at the stage of registering and provisioning nodes. The Cornell Aristotle infrastructure team has been advising Dartmouth on the OpenStack cloud implementation, a clear benefit of being a member of a federation. Dartmouth is awaiting approval from the Provost's office for their Ceph cluster. Dartmouth hardware and software are self-funded.
- Excellent presentation by a Cornell REU student Jackie Zheng who is working on getting WRF and the OpenMORDM analytic framework running and bringing up MPI clusters in cloud. The radio astronomy REU student is focused on machine learning.
- When getting containers to talk to each other, OpenStack has mysterious networking issues which we are working on to overcome.
- XDMoD data was pulled into the Cornell database for the first time in July; this process will be automated next.
- UB is working on kernel containerization with Docker and Singularity in some cases, benchmarking initially on their old OpenStack instance and also directly on the node controller to get a comparison. Next step will be to get these running on Aristotle at UCSB and Cornell. They will look at the RightScale API after that.
- Aristotle consultant Brandon Barker is making good progress running the Angela Douglas metabolic mode of gut microbiota application on 28 cores.

8/21/2019 status meeting:

- Continuing to prepare for Aristotle and public cloud benchmarking of linear algebra, radio astronomy Fast Radio Burst pipeline (AWS spot/S3 workflow), OpenMORDM Lake_Problem_DPS/WaterPaths, and WRF (AWS batch and parallel). Plan to include network performance.
- WaterPaths is running on a 28 core VM in a container using all cores and is ready to scale.
- Users want on demand clusters they can make or destroy in minutes.

- The Exotanium team (http://revithaca.com/exotanium-inc-is-fortifying-the-cloud-computing-revolution/) wants to publish running WRF fast. They are looking for data that would help them develop X-Containers and make them better. Goal is to provide containers with less overhead and more security.

8/28/2019 status call:

- Completed comparative benchmarking on Red Cloud and Azure for a general Terraform and Ansible cluster.
- Some billing tags were patched so we can see things in RightScale. Billing 24 hours rather than real-time is an issue.
- Work continues on DrAFTS 2.0 price comparison tool.
- A group of students are growing their own food for campus at UCSB (Edible Campus project) and the Aristotle team will be instrumenting it.
- UB hardware order went in this week. They continue to struggle with the automation of OpenStack and how to script OpenStack in a way that the UB security team will accept.
- Dartmouth continues to work on auto provisioning their nodes with the TripleO configuration that Red Hat and CentOS use by default. UCSB senior systems administrator Jeff Oakes has done extensive work (it is challenging) with TripleO and will assist Dartmouth.
- We are getting usage data from all 3 sites and core usage is now showing on the portal pages.

8/28/2019 status meeting:

- Aristotle "how to" documentation is being tested by Dartmouth as they deploy their first OpenStack cloud and "build your own federation" scripts are being continuously improved.
- We share our accounting data through a JSON file and are working on improving its security.
- Portal is almost completed; we may pull in some more Open XDMoD graphics.
- Plans are underway to share all code from the project on GitHub during PY5 (database schema, templates for the portal, etc.).
- We had cross federation runs of CENTAURUS by manually bringing the application up at UCSB and spilling over to Cornell when more workers were needed. Unlike Eucalyptus, the OpenStack API doesn't allow auto scaling across the federation. Containerization is our main focus for migrating applications to campus, NSF, and public clouds.
- Dartmouth is running Kubernetes on Cornell's Red Cloud for a new use case focused on using Jupyter Notebooks in the cloud rather than physical computer labs.
- API work is needed on RightScale. Goal is a comparative page on the portal for some sample apps.

9/4/2019 status meeting:

- Plan to use public IP rather than private network for Terraform and Ansible automated cluster creation benchmarking.
- Meeting planned with RightScale technical team to discuss the RightScale Optima dashboard capabilities. At the same time, we will be looking at Google Cloud and AWS tagging labels. Google Cloud usage should come into the billing center daily.
- Cornell's Chris Myers will be developing a process to use AWS spot instances and to optimize usage.
- We'll be comparing run times between Red Cloud and select public clouds for use case applications; will need to know starting parameters, data size, size of first instance, time to execute, etc.

- Plan to run WRF with AWS ParallelCluster; it's still being tested and there are lots of issues. Will be conferring with the Cornell Cloudification team and AWS HPC team. If we can run ParallelCluster, we will have one dedicated small instance for the master head node and schedule spot instances with it. Amazon Elastic Computer Cloud has very fast network bandwidth but spot instances cannot be scheduled with it. There are cost and performance tradeoffs here.

2.0 DIBBs Acquisition, Installation, Configuration, Testing & Maintenance Report

2.1 Hardware Acquisition

- No purchases were made this quarter.

2.2 Installation, Configuration, and Testing

- Cornell tested their disaster recovery plans after a building power outage on August 9th took down Red Cloud and the Ceph Storage cluster. All hardware survived and was back in production by the close of business on August 10th. Other work this quarter involved testing the GPU node with various images (Linux and Windows) and assisting Dartmouth with OpenStack.
- UB focused on two areas this quarter: GPU support and Ceph upgrades. For GPU support, Cornell shared the Puppet code they used to allow the V100 GPU to run in passthrough mode on instances. This was implemented on a NVIDIA T4 system in the development cloud to run benchmarks and verify performance. Given the success with the T4, they are in the process of procuring a V100 system for the production cloud. For Ceph upgrades, the four new OSD servers purchased last quarter have been added to the current Ceph pool. Next, UB will take out one existing server at a time and replace the mixed-use SSDs for write intensive SSDs because they feel they are much better suited for that environment. They are also replacing the Broadcom 10GbT NICs that came with the systems with Intel 10GbT cards because they have had issues with the NIC/Linux Driver causing the network card to flake out and have to reboot to recover. They have seen this on multiple machines.
- UCSB updated their Ceph storage, attempt to update OpenStack (stalled with Mistral bug), and helped Dartmouth with TripleO installation of OpenStack.
- Dartmouth collaborated with Cornell and UCSB to accelerate resolution of deployment bottlenecks. They worked through node registrations issues and are back on track. They are optimistic that they will have their OpenStack instance up and running in by October 1st.

2.3 Federated Identity Management

Researchers use single sign-on at any member site.

2.4 Cloud Status by Site

The chart below shows each site's production cloud status. Dartmouth's is still in progress.

	Cornell	Buffalo	UCSB
Cloud URL	https://redcloud.cac.cornell.edu	https://lakeeffect.ccr.buffalo.edu/ (access only to federation)	https://openstack.aristotle.ucsb.edu/
Status	Production	Production	Production
Software Stack	OpenStack	OpenStack	OpenStack
Hardware Vendors	Dell	Dell, Ace	Dell, HPE, DXC
DIBBs Purchased Cores	*616	**256	356
RAM/Core	8GB	up to 8GB	9GB Dell, 10GB HPE
Storage	Ceph (1392TB)	Ceph (720TB)	Ceph (528TB)
10gb Interconnect	Yes	Yes	Yes
Largest instance type	28core/240GB RAM	24core/192GB RAM	48core/119GB RAM
Globus File Transfer	Yes	Yes	Planned
Globus OAuth 2.0	Yes	Yes	Yes
Total Cores (DIBBs purchased cores + existing cores) = 2060	* 616 additional cores augmenting the existing Red Cloud (1064 total cores).	** 256 additional cores augmenting the existing Lake Effect Cloud (424 total cores).	***356 cores in UCSB Aristotle cloud (572 total cores, Aristotle is separate from UCSB campus cloud)

2.5 Tools

- Red Hat OpenStack – Cornell, UB, and UCSB all have production OpenStack clouds. Dartmouth's is in progress.

3.0 Cloud Federation Portal Report

Content updates to the project portal are ongoing (https://federatedcloud.org).

Open XDMoD continues to monitor data ingestion from all sites, as well as provide the utilization data (https://federatedcloud.org/using/federationstatus.php).

The portal planning table was not updated this quarter:

Portal Framework			
Phase 1	**Phase 2**	**Phase 3**	**Phase 4**
10/2015 – 3/2016	**4/2016 – 12/2016**	**1/2017 - End**	**1/2017 - End**
Gather portal requirements, including software requirements, metrics, allocations, and accounting. Install web site software.	Implement content/functionality as shown in following sections. Add page hit tracking with Google Analytics, as well as writing any site	Implement content/functionality as shown in following sections. Add additional information/tools as needed, such as selecting where to run based on	Release portal template via GitHub. Update periodically.

	downloads to the database.	software/hardware needs and availability.	
Documentation			
Phase 1	**Phase 2**	**Phase 3**	**Phase 4**
10/2015 – 3/2016	**4/2016 – 10/2016**	**11/2016 – End**	**1/2017 - End**
Basic user docs, focused on getting started. Draw from existing materials. Available through CU doc pages.	Update materials to be federation-specific and move to portal access.	Add more advanced topics as needed and after implementation in Science Use Cases, including documents on "Best Practices" and "Lessons Learned." Check and update docs periodically, based on ongoing collection of user feedback	Release documents via GitHub. Update periodically.
Training			
Phase 1	**Phase 2**	**Phase 3**	**Phase 4**
10/2015 – 3/2016	**4/2016 – 12/2017**	**4/2017 – 12/2017**	**1/2018 - End**
Cross-training expertise across the Aristotle team via calls and science group visits.	Hold training for local researchers. Offer Webinar for remote researchers. Use recording/materials to provide asynchronous training on the portal.	Add more advanced topics as needed. Check and update materials periodically, based on training feedback and new functionality.	Release training materials via GitHub. Update periodically.
User Authorization and Keys			
Phase 1	**Phase 2**	**Phase 3**	**Phase 4**
10/2015 – 1/2016	**2/2016 – 5/2016**	**6/2016 – 3/2017**	**4/2017 – End**
Plan how to achieve seamless login and key transfer from portal to Euca dashboard.	Login to the portal using InCommon.	Beta testing Euca 4.4 with Euca console supporting Globus Auth. Will deploy and transition to Euca 4.4 on new Ceph-based cloud.	Transition to OpenStack console with Globus Auth login.
Euca Tools			
Phase 1	**Phase 2**	**Phase 3**	**Phase 4**
10/2015 – 3/2016	**4/2016 – 12/2016**	**1/2017 – End**	**1/2017 – End**
Establish requirements, plan implementation.	No longer relevant since Globus Auth will let us interface with Euca web console	N/A	N/A
Allocations and Accounting			
Phase 1	**Phase 2**	**Phase 3**	**Phase 4**
10/2015 – 3/2017	**3/2017 –5/2018**	**6/2017 – 10/2018**	**6/2017 – End**

#1541215

Plan requirements and use cases for allocations and account data collection across the federation. Design database schema for Users, Projects and collections of CPU usage and Storage Usage of the federated cloud.	Display usage and CPU hours by account or project on the portal. Integration hooks for user and project creation/deletion and synchronization across sites. Note: due to OpenStack move, account creation across sites is delayed.	Automate project (account) creation by researcher, via the portal.	Report on usage by account, if the researcher has multiple funding sources. Release database schema via GitHub.

3.1 Software Requirements & Portal Platform

No software changes were made to the portal platform this quarter.

3.2 Integrating Open XDMoD and DrAFTS into the Portal

A UCSB REU student joined the science team this summer to help develop a new technology stack for Aristotle. UCSB is currently recruiting a new student to work on the DrAFTS 2.0 price prediction tool; the plan is to execute the tool in October with a target delivery date of December 2019.
.
The XDMoD team is preparing for the release of Open XDMoD 8.5, which will include a beta version of the Novice User Dashboard and a number of minor bug fixes in cloud metrics. The team has also provided Cornell with an initial REST API to support the extraction of cloud accounting data from XDMoD for incorporation into the Aristotle portal. This data is now used to track federated cloud usage on the Federation Status page. The XDMoD team will continue to iterate with the Cornell to make improvements to this process.

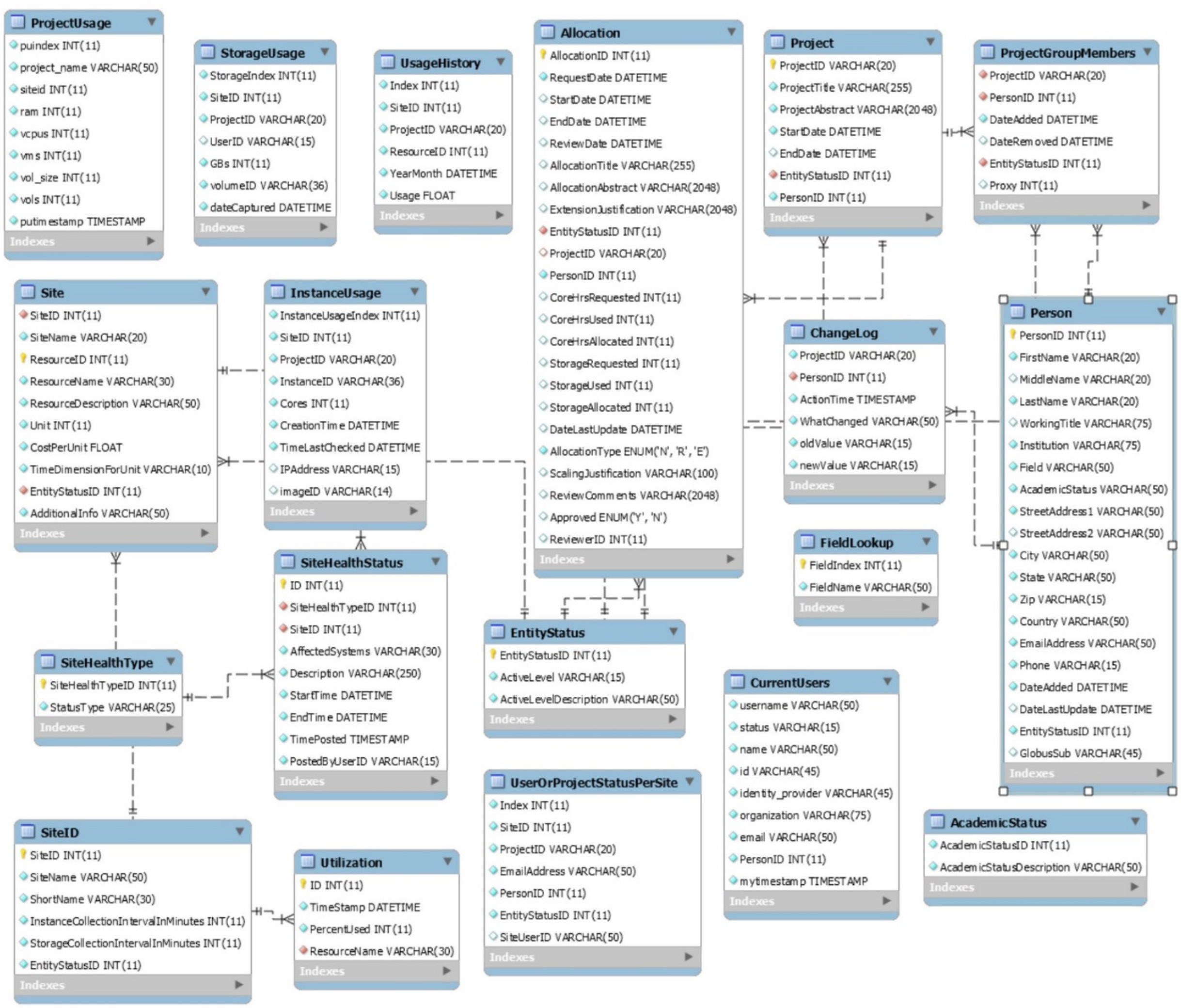
Progress has been made containerizing XDMoD application kernels so that they may be more easily and consistently deployed on cloud infrastructures. The XDMoD team plans to begin deployment of application kernels on all Aristotle sites in the coming quarter. Application kernels (AK) are short benchmarks based on the actual real-world applications and benchmarks. They are used for performance monitoring as well as benchmarking. One of the difficulties in AK use is the need to compile them on each monitored resource. AK building can be a lengthy process due to dependencies on multiple libraries. Containerization will simplify the installation process and allow rapid deployment of AK for performance monitoring purposes.

During the last quarter, we made an initial approach for AK containerization and improved the OpenStack integration into Application Kernel Remote Runner. The initial containerization was done for HPCC, HPCG, NWChem, NAMD, IOR, and MDTest AKs. An additional effort was made to ensure that created Docker containers can also work with Singularity, allowing execution of AK containers in traditional HPC environments. HPCC, HPCG, NWChem, and NAMD are ready for the next stage of making production containers. The containerized AK automatically detects a number of CPU cores and associated input parameters for this job size. This was a containerized AK that can be easily executed on most platforms without the need to specify core count or input problem. There are still some unresolved issues with IOR and MDTest due to the difficulties of purging disk caches to obtain realistic file system performance.

During the initial approach, AKs were compiled for generic CPU architectures. For the majority of compute-intensive applications, an architecture-specific optimization can give a significant boost in performance. Generic, common denominator, CPU type OpenStack allows exposure of complete CPU instructions to the instance allowing execution of architecture optimized binaries. The production containers would include multiple optimized binaries. The proper executable will be automatically selected based on the detected architecture. Currently, HPCC AK has reached the pre-production stage and needs further testing on deployment targets. Future work includes: finalizing containers for HPCG, NWChem, and NAMD AKs, identifying ways to reliably measure file system performance, and deploying AK on Aristotle for performance monitoring and comparison between different sites.

3.3 Allocations & Accounting

No changes were made to the database schema this quarter:



4.0 Research Team Support

4.1 Science Use Case Team Updates

New Aristotle publications for 2019 not previously reported include:

- Bakir, F., Wolski, R., Krintz, C., and Ramachandran, G. (2019). Devices-as-services: Rethinking scalable service architectures for the internet of things. *Proceedings of the 2019 USENIX Conference*. https://www.usenix.org/conference/hotedge19/presentation/bakir
- Seneca: Fast and low cost hyperparameter search for machine learning models, *IEEE Cloud 2019.* https://sites.cs.ucsb.edu/~ckrintz/papers/ieee_cloud19.pdf
- Carson, K., Thomason, J., Wolski, R., Krintz, C., and Mock, M. (2019). Mandrake: Implementing durability for edge clouds. *IEEE International Conference on Edge Computing.* https://sites.cs.ucsb.edu/~rich/publications/edge-19.pdf
- Lin, W.T., Bakir, F., Krintz, C., and Mock, M. (2019). Data repair for distributed, event-based IoT applications. *Proceedings of the 13th ACM Conference on Distributed and Event-based Systems.* https://dl.acm.org/citation.cfm?id=3328905.3329511
- George, G., Wolski, R., Krintz, C., and Brevik, J. (2019). Analyzing AWS spot instance pricing. *2019 IEEE International Conference on Cloud Engineering (IC2E).* https://ieeexplore.ieee.org/document/8790118
- Letson, F., Barthelmie, R.J., and Pryor, S.C. (2019). RADAR-derived precipitation climatology for wind turbine blade leading edge erosion. *Wind Energy Science*. https://www.wind-energ-sci-discuss.net/wes-2019-43/wes-2019-43.pdf.

Cornell Aristotle use case support team members Bennett Wineholt and Pete Vaillancourt provisioned multiple VM MPI clusters on demand in Google Cloud and AWS. Work continues to fully translate Terraform and Ansible automation to Microsoft Azure and Cornell Red Cloud's OpenStack. Multiple small VM MPI tests have been executed and Docker versions of OpenMORDM, WRF, and Lake Problem test cases are being developed. We will evaluate performance of these tests based on the job runtimes, hardware utilized, and cost.

REU student Jackie Zheng produced a script which sets up Google and AWS instances to run the Lake Problem on different instances. Originally, the Lake Problem repository created a Docker container cluster in a single instance. Jackie used Ansible and Terraform to configure the Docker containers to allow for multi-instance communication where each instance takes up a single Docker container. Jackie also created a Docker container which will run the scripts inside a container so that the host does not need additional dependencies. Jackie collaborated with Brandon Barker, Pete Vaillancourt, and Bennett Wineholt.

Use Case 1: A Cloud-Based Framework for Visualization & Analysis of Big Geospatial Data
We are currently developing an iGlobe-based application to support sustainability research at UB. The plan is to create a data and computing infrastructure, powered by Aristotle, to support a large collection of researchers, community stakeholders, sustainability and resiliency planners, working towards creating partnerships for sustainable development of the Lower Great Lakes area. During the inaugural NSF-funded planning workshop, held in August 2019, we demonstrated the first version of the Aristotle platform. Next steps are to refine the platform and allow researchers to contribute data and run analyses.

Use Case 2: Global Market Efficiency Impact

We continued to investigate how price deviations affect liquidity using the financial framework that runs on Aristotle (Reichers). We will complete and resubmit a publication to the *Journal of Financial Economics* next quarter.

Use Case 3: Application of the Weather Research and Forecasting (WRF) Model for Climate-Relevant Simulations on the Cloud

Précis objectives of our current suite of simulations:

1. Quantify impact of resolution (to convective permitting scales) on near-surface flow (i.e., wind speed) regime fidelity
2. Examine scales of coherence in wind fields. Specifically, spatial scales of calms (i.e., wind speeds < 4 m/s), and spatial scales of intense wind speeds (i.e., wind speeds > the local 90th percentile value)
3. Quantify the platform dependence of wind simulations (i.e., quantify the differences in near-surface wind regimes from simulations conducted on conventional HPC and the cloud)
4. Examine inter-annual variability in near-surface wind speeds (can we simulate it, what is the source?)
5. Evaluate impact of large wind turbine (WT) developments on downstream climate (local to mesoscale).

We are addressing these objectives by conducting and analyzing the output from high-resolution numerical simulations with the Weather and Research Forecasting model (WRF, v3.8.1).

As mentioned previously, activities on Aristotle were stopped in the 2nd quarter of 2019 due to the urgent need to transition to OpenStack. Thus, the simulations with WRF we were conducting to assess the resolution sensitivity had to be terminated after 9-months of simulation instead of the 12-months as originally planned. Nevertheless, subsequently it was possible to conduct a series of short additional experiments to examine output sensitivity to the precise way in which the simulations are performed. The resulting simulation output provides key information regarding the performance and sensitivity of key parameterizations within the WRF model to resolution. Analyses of this, and other output from our simulations on Aristotle, is being conducted on our Jetstream allocation and is the focus of a journal manuscript that was submitted by Tristan Shepherd in May to the *Journal of Applied Meteorology and Climatology* and was also the subject of conference presentations during May and June at the *Wakes Conference* and the *Wind Energy Science Conference* by Sara C. Pryor. Analysis conducted of the most recent WRF output during this quarter by Pryor is the focus of a manuscript recently submitted to the journal *Wind Energy Science*.

We recruited an undergraduate student to participate in our science case. The student's skill set turned out to be not the best match for this project but, nonetheless, he was undoubtedly enriched by the experience.

In preparing output and sample MATLAB code for the REU student to use in their research, Pryor great progress towards addressing a key science research question regarding differential credibility of numerically derived extreme wind speed estimates (Objective 4). The concept of differential credibility is an emerging topic in atmospheric science. In brief, the objective of Pryor's research in this arena can be paraphrased as follows: Is it possible to quantify how differential credibility of any climate projection (e.g. of the 50 year return period wind speed) depends on factors such as the dynamical causes of, and context for, extreme wind speeds and thus the likelihood that such features are reproduced with fidelity for a given model grid formulation/spacing/duration? In other words, are climate projections of this parameter differentially credible in space and/or with model resolution/formulation? Pryor recently submitted a short manuscript on this new approach to assessing differential credibility to the *Journal of Physics Conference* series and will present this research at a conference in October.

Activities planned for next quarter:

- Our activities will focus on additional WRF simulations and analyses of WRF output generate to data in support of Objective 4 and 5.

Journal manuscripts:

- Pryor, S.C., Shepherd, T.J., Barthelmie, R.J., Hahmann, A. Volker, P. (2019). Wind farm wakes simulated using WRF. *Journal of Physics: Conference Series (in press).*

Presentations:

- Pryor, S.C. (2019). Sensitivity of array-array interactions to wind farm parameterization and resolution: Meteorology measurement and modeling for offshore and coastal and land related to wind farms colloquium. *DTU Wind Energy Dept.*, Roskilde, Denmark, July 2019 *(invited presentation).*

Forthcoming conference presentations:

- Pryor, S.C., Shepherd, T.J., Bukovsky, M., and Barthelmie, R.J. (2019). Assessing the stability of wind resources and operating conditions. *North American Wind Energy Academy Wind/Tech Conference*, Amherst, MA, October 2019 *(oral presentation).*
- Letson, F. Shepherd, T.J., Barthelmie, R.J., and Pryor, S.C. (2019). Modelling hail and convective storms with WRF for wind energy applications. *North American Wind Energy Academy Wind/Tech Conference*, Amherst, MA, October 2019 *(oral presentation).*
- Shepherd, T.J., Barthelmie, R.J., and Pryor, S.C. (2019). Assessment of wind turbine impact on future climate in GCM-driven WRF simulations. *North American Wind Energy Academy Wind/Tech Conference*, Amherst, MA, October 2019 *(oral presentation).*
- Shepherd, T.J., Barthelmie, R.J., and Pryor, S.C. (2019). Quantifying array-array effects using WRF model simulations: A sensitivity analysis. *North American Wind Energy Academy Wind/Tech Conference*, Amherst, MA, October 2019 *(oral presentation).*

Use Case 4: Transient Detection in Radio Astronomy Search Data

We continued to work on the Fast Radio Burst (FRB) Pipeline to implement parallelization techniques that speed up the processing of large datasets. We have implemented a method of breaking up the data (dynamic spectra) into chunks so that analysis can be done on multiple chunks at a time. We have verified this method is successful in detecting FRBs, depending on how the data is split up, for the Friends-Of-Friends (FOF) algorithm in serial form. We are currently implementing this method in parallel using multiprocessing on the same machine. Since the underlying method is generalizable, we are hoping to use it with other algorithms in the future. Furthermore, we continue to add the features necessary to make the pipeline interruptible (checkpoint/restart) with logging so that it can be run in parallel on multiple AWS Spot Instances. These test runs will also help in the evaluation of the RightScale Optima platform.

In addition, we have been developing containers for easy deployment of the pipeline. One of these containers is a public version of the previous development container (based on a NANOGrav container)

NSF
#1541215

which contains all the relevant pulsar and radio transient detection software. The other will be a reduced-size and more bare-bones version that anyone can deploy on any system which has Docker installed to run analysis on any radio astronomy data using our pipeline. We will use this smaller version on AWS. We have also begun work to make the pipeline more general than just FRB detection in order to encompass methods used in pulsar astronomy and other radio transient detections.

REU student Ryan Hill built a functioning neural network that classifies different types of radio frequency interference into one of 5 categories, including no RFI. It does so with up to 90% accuracy depending on the loss rate. Currently, Ryan is training over many different loss rates trying to identify patterns in the misclassifications. Depending on what is found, he will modify the network, possibly adding additional layers, then will extend it to classify images that include single pulses and multiple different types of RFI to see how it responds. The code for the neural network exists in a Jupyter Notebook and .py versions; in addition, Ryan created code to develop, use, and modify the RFI and single pulse graphs.

Use Case 5: Water Resource Management Using OpenMORDM
The Cornell Aristotle team modified an existing Docker container to create a VM for initial tests with WaterPaths, which automatically clones and compiles WaterPaths and Borg, and downloads all necessary data files. On this test VM, WaterPaths was successfully adapted to run on Red Cloud, after some issues with binary data formats were resolved, which allowed for standalone WaterPaths simulations and small-scale optimization runs with 4 MPI processes (one master and three slaves). Lastly, the Aristotle team and Patrick Reed research group defined the guidelines for the next steps, which will include timings and MPI profiling:

- Using the same size VM with 28 cores, set up 3 Docker containers to use 8 cores each (total of 24 cores)
- Set up a small cluster of 3 VMs, each with 8 cores, with a single Docker container on each that uses all 8 cores
- Set up a larger cluster of 8 VMs, each with 28 cores, with a single Docker container on each VM.

The timings and MPI profiling are expected to result in at least two publications, one focused on WaterPaths and the other on the cloud itself.

Use Case 6: Mapping Transcriptome Data to Metabolic Models of Gut Microbiota
We fine-tuned our analysis using SteadyCom to investigate interactions occurring among microbial members of a Drosophila gut community. We had previously applied arbitrary minimum growth rate constraints to all microbial community members to ensure that all microbes were capable of growth as observed *in vivo*. These constraints, while technically sound, generated optimal growth flux values that were not 'central,' were sensitive to solver type and version used, and displayed flux vectors with futile cycles. To mitigate these issues, we adopted a new approach where we use Flux Variability Analysis (FVA) and medoid growth rate constraints to obtain 'central' solutions. We are also using L1 minimized fluxes to remove futile cycles and otherwise extraneous flux predictions. Application of these new constraints has yielded more desirable flux vectors with no futile cycles and we plan to adopt these parameters as our default method for analyzing gut microbial community interactions. The new simulation methodology required more computational resources due to the use of FVA to obtain a large set of fluxes for calculating the medoid. We used Aristotle's largest instance size (28 cores) to mitigate this, and noticed only minor increases in runtime due to the SteadyCom FVA implementation being parallel.

NSF
#1541215

Use Case 7: Multi-Sourced Data Analytics to Improve Food Production & Security

Citrus Under Protective Screening (CUPS) citrus greening project:
Originally scheduled for July, construction was delayed and the CUPS facility is now scheduled to break ground in September. The science team is preparing what it can in the lab prior to deployment, but installation and testing must wait until the structure is in place and construction safety restrictions can permit the team to visit and work onsite. The Aristotle team will attend the infrastructure repair visit meeting next month to see if a kick-off date for the installation can be determined.

REU student Gareth George developed an AWS compatibility layer for microcontrollers to use as part of the CUPS project. Specifically, AWS Lambda is a facility that is popular with the science team but restricted for use to AWS. This student (acting on his own initiative) has created a compatibility service that not only runs in Aristotle, but can also run on resource restricted devices themselves. From a systems perspective, this is a significant breakthrough and we are now preparing a paper for publication on the service.

Citrus frost prevention (Lindcove Research and Extension Center, Exeter, CA):
We are continuing to harden the infrastructure. While one of the principle hardware components did fail this summer, the backup and redundancy engineering implemented last winter kept the full system operational. A repair visit is scheduled for next month to replace the failed hardware and also to plan for the first field test of the system that will occur this frost season (November through March), weather permitting.

Edible Campus Project
The Edible Campus Project (http://www.sustainability.ucsb.edu/ediblecampus/) has contacted the Aristotle science team about using Chandra Krintz's SmartFarm technologies (used in frost prevention and CUPS) to monitor Edible Campus farm sites. REU student Gabriel Soule has become the primary developer and deployment engineer for this collaboration. Currently, he is siting solar power and long-distance Wi-Fi at an Edible Campus farm located on unincorporated land next to UCSB.

5.0 Community Outreach and Education

5.1 Community Outreach

- Cornell's Rich Knepper gave an invited presentation at *PEARC19* on "Red Cloud and Aristotle: campus clouds and federations;" Knepper discussed lessons learned in helping researchers leverage these resources as well as leveraging other research cloud infrastructure and transitioning to public cloud.
- Knepper, Resa Reynolds, Eric Coulter, and Steve Bird also led a *PEARC19* BOF on "Campus Research Clouds."

5.2 Education

- Dartmouth's first use case is focused on using Jupyter notebooks and the cloud for education to reduce the number of campus computer labs which most of the time aren't being used; Dartmouth is using Cornell's Red Cloud for development.

NSF
#1541215