ARISTOTLE
CLOUD FEDERATION

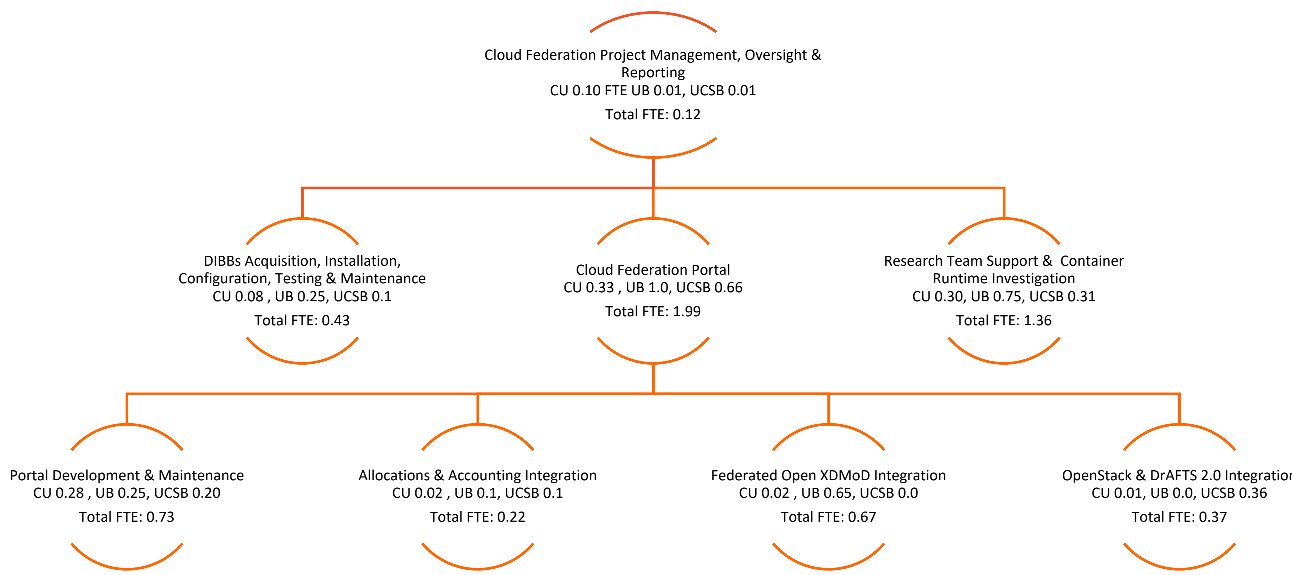
CC*DNI DIBBs: Data Analysis and Management Building Blocks for Multi-Campus Cyberinfrastructure through Cloud Federation

Program Year 6: Quarterly Report 4
(No-Cost Extension Year Ending 9/30/2021)

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

9/27/2021

This is the Program Year 6: Quarterly Report 4 (PY6 Q4) of the Aristotle Cloud Federation team. This report is the last quarter of a one-year No-Cost Extension. 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 ... 3
1.1 Subcontracts ... 3
1.2 Project Change Request ... 3
1.3 Project Execution Plan ... 3
1.4 Project Planning Meetings ... 3

2.0 Container Runtimes Investigation Report ... 4

3.0 DIBBs Acquisition, Installation, Configuration, Testing & Maintenance Report ... 5
3.1 Hardware Acquisition ... 5
3.2 Installation, Configuration, and Testing ... 5
3.3 Federated Identity Management ... 6
3.4 Cloud Status by Site ... 6

4.0 Cloud Federation Portal Report ... 6
4.1 Software Requirements & Portal Platform ... 6
4.2 Integrating DrAFTS into the Portal ... 7
4.3 Integrating Open XDMoD into the Portal ... 7
4.3.1 Containerized Application Kernels (AK) on Google Cloud ... 7
4.3.2 XDMoD Cloud Integration ... 11
4.4 Allocations & Accounting ... 11

5.0 Research Team Support ... 11
5.1 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 ... 11
... 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 ... 14

6.0 Community Outreach and Education ... 15
6.1 Community Outreach ... 15
6.2 Education ... 16

7.0 Future Plans ... 17

1.0 Cloud Federation Project Management, Oversight & Reporting

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

1.2 Project Change Request
No change requests.

1.3 Project Execution Plan
The Project Execution Plan (PEP) was approved by NSF on 12/18/2015. We operated as planned.

1.4 Project Planning Meetings

7/2/2021 status call:

- Working with Aristotle computational scientist Peter Vaillancourt, Cornell undergraduate Jeffrey Lantz began documenting his experiences running a new, large-scale WRF application, containerized with Intel compilers and MPI, over multiple VMs in public clouds. The goal is to use 16 or more cloud VMs that match Stampede2 VMs as closely as possible. Lantz and Vaillancourt are also exploring the feasibility and cost of various storage options based upon ~20TB data output. Google Cloud storage options are Aristotle Red Cloud NSF Server, In-Cluster NSF Server, Filestore, NetApp Cloud Volumes Services (CVS), and Cloud Store with Fuse; AWS options are In-Cluster NFS Server on EBS, EBS Multi-Attach, Amazon EFS CSI Driver, and Amazon S3; and Azure options are In-Cluster NSF Server on Azure Disks and Azure Files.
- The container runtime investigation team reinstalled Singularity for additional space, investigated ways to capture performance metrics (VTune, confab and Slurm accounting on Stampede2), and met with Intel to discuss MPI issues experienced while running MPI in a WRF container. Next, the team will test Singularity on Stampede2, work on multi-VM MPI, test the WRF container in X-Containers, and work on automation issues, including AWS permissions, Terraform for network creation, and IAM roles.

7/6/2021 status call:

- The container runtime investigation team plans to write a technical report on the challenges users face when using containers for scientific applications and provide lessons learned to the wider community. The team also plans to participate in a SC21 tutorial on containers. The report will provide an overview of container technology, use case descriptions, some performance experience, application orchestration strategies, and considerations when selecting and building containers.

7/15/2021 status call:

- Our AWS cluster with Terraform is running with core functionality and benchmarks. We will try to run on Google Cloud and Azure next.
- WRF is running with Open MPI and we're working on implementing a new application for the Sara C. Pryor group.
- The Intel VTune version on Stampede2 is different from the VTune version inside our container; we're discussing this and other issues with Intel.
- We continue to document the pros/cons for a variety of container strategies and are capturing development time, time in queue, and other insights to share with the broader community.

7/20/2021 status call:

- Our cluster deployments in AWS and Google Cloud are running, and we're testing WRF runs in AWS with volume attached to improve I/O. The runs are scripted, including bare metal scripts. Our Singularity container needs a bit more testing on Stampede2 and we need to test our Slurm virtual cluster in Red Cloud. We're documenting netCDF gotchas for inclusion in the technical report.
- We presented our container experiences to date at a *PEARC'21* tutorial. Our work with WRF is ideal for ferreting out why containers are difficult to build and deploy for the cloud as WRF hits all the hot spots.
- Our novel work with Kubernetes is proceeding and we're documenting our experiences as we go. Overall, we're focused on understanding how difficult it is for a new user to use orchestration.

8/24/2021 status call

- We ran the new Intel-compiled WRF science code on 36 cores for 5 days on Aristotle and ran multi-node (4 nodes, 48 cores each) benchmarks on AWS. We'll run the Singularity container next. Bare metal numbers are currently under review.
- We're focused on comparing AWS (with its 10 Gigabit Ethernet) vs. bare metal Stampede2 runs using comparable memory size and processor features. We want to compare performance to get a better understanding of the suitability of AWS virtualized hardware for running science codes. We believe AWS performance will be good and Stampede2 performance will be better. The question will then be how much better, and given that data point, what is the penalty or relative trade-off of running on AWS when you consider you don't have to sit in a queue.
- We're also finalizing scripts and Terraform Kubernetes set-ups for community use.

2.0 Container Runtimes Investigation Report

We completed our container investigation and the "Aristotle Cloud Federation: Container Runtimes Technical Report" is now available:
https://federatedcloud.org/papers/ContainerRuntimesTechReport.pdf.

The report identifies pain points experienced by users when selecting and using Docker, Singularity, and X-Containers and documents lessons learned. Experiences with Kubernetes and orchestration on campus and public clouds are included. Knowledge gained will be shared during the "A Deep Dive into Constructing Containers for Scientific Computing and Gateways'' tutorial at the SC21 Conference.
Technologies and techniques developed during this investigation that are now available to the wider community include:

- **Automated Deployment Methods** – implemented a Slurm HPC cluster in a cloud with OpenHPC 2 series based on CentOS/Rocky Linux 8.
  https://github.com/federatedcloud/wrf-cluster-openstack
- **Metabolic Model and Container** – developed code to predict metabolic function of the gut microbiota of Drosophila melanogaster using v.3.0.4 of the OpenCOBRA Toolbox and v7.51.1 of the Gurobi Optimizer. An optional, containerized environment for running the code is available as well as a tutorial for performing the simulations.
  https://github.com/federatedcloud/DouglasMetabolicModels/releases/tag/v1.0.2
  https://github.com/federatedcloud/COBRAContainers

#1541215

- **Radio Astronomy Container** – developed a single container of radio astronomy software that combines the pipeline components developed for pulsar and other transient detections that can be deployed either on the cloud with Docker or on an XSEDE HPC resource with Singularity. https://github.com/federatedcloud/pulsar-pipeline-container
https://hub.docker.com/r/cornellcac/pulsar-pipeline
https://datasets.datalad.org/?dir=/shub/federatedcloud/pulsar-pipeline-container
- **Kubernetes Implementation Code for MPI Clusters** – this code reflects the current state of experimental support for MPI applications managed by Terraform Kubernetes constructs that allow for automatic node count scaling and cloud portability. The software-based resource provisioning can best be attempted on AWS at the present time: it also works with Google Cloud with MPI applications. Conversion to other cloud platforms should be possible with extensive changes to a platform-specific Terraform Kubernetes provider or other resource configurations. Familiarity with Kubernetes software concepts, resources provisioning on the desired cloud, and debugging parallel computing applications are recommended. The repository includes a "Getting Started with Kubernetes" tutorial as well.
https://github.com/federatedcloud/kubernetes-mpi-clusters
- **Sparta** – in order to protect edge clouds from overheating, we developed a heat-budget-based scheduling system called Sparta which leverage dynamic voltage and frequency scaling to adaptively control CPU temperature. Sparta takes machine learning applications, datasets, and a temperature threshold as input. It sets the initial frequency of the CPU based on historical data and then dynamically updates it, according to the applications' execution profile and ambient temperature, to safeguard edge devices.
https://sites.cs.ucsb.edu/~ckrintz/papers/edge21.pdf
- **WRF CONUS Benchmark Containers** – implemented WRF 4.2.2 to run CONUS (Continental US) benchmarks on bare metal HPC in a Docker and a Singularity container.
https://github.com/federatedcloud/WRFv4-Benchmarking
- **WRF Docker Container** – implemented a Docker container for WRF 3.8.1 with a Fitch patch.
https://github.com/federatedcloud/Docker-WRF-3.8.1-Fitch

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

3.1 Hardware Acquisition

- Cornell had no hardware acquisitions this quarter.
- UCSB acquired 2 GPU nodes (Dell C4140).
- UB acquired 4 standard cloud nodes (Dell R650), 2 large memory cloud nodes (Dell 650), 2 GPU nodes (Dell R750), and 1 CEPH OSD node (Dell R740xd).

3.2 Installation, Configuration, and Testing

- Cornell installed 4 additional cloud nodes (purchased with Cornell funds), adding 112 physical cores to Red Cloud.
- UCSB installed 2 GPU nodes, adding 88 physical cores and 8 NVIDIA Tesla V100 GPUs to their Aristotle cloud.

#1541215

- UB installed 8 cloud nodes and 1 storage node, adding 320 physical cores, 4 NVIDIA A100 GPUs, and 96TB storage capacity to CEPH. In addition, they completely migrated off Red Hat OpenStack 13 and onto OpenStack Wallaby, allowing them to implement OpenStack Magnum and Octavia Load Balancing Support. With the addition of the NVIDIA AMD GPUs, they can now support virtual GPU (vGPU).

3.3 Federated Identity Management

Researchers use single sign-on at any member site.

3.4 Cloud Status by Site

	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	**792	***740
RAM/Core	8GB	up to 8GB	9GB Dell, 10GB HPE
Storage	Ceph (1.6PB)	Ceph (864TB)	Ceph (720TB)
10gb Interconnect	Yes	Yes	Yes
Largest Instance Type	28core/192GB RAM	24core/192GB RAM	48core/119GB RAM
GPUs	24 GPUs	13 GPUs	8 GPUs
Globus File Transfer/End Points	Yes/CAC Home Directories	Yes/CCR Home and Projects Directories	Yes/POSIX UCSB Aristotle
Globus OAuth 2.0	Yes	Yes	Yes
Total Cores (DIBBs purchased cores + existing cores) = 3280	* 616 additional cores augmenting the existing Red Cloud (1428 total cores).	** 1112 total cores	***740 cores in UCSB Aristotle cloud (956 total cores, Aristotle is separate from UCSB campus cloud)

4.0 Cloud Federation Portal Report

The portal was updated throughout the quarter.
https://federatedcloud.org

4.1 Software Requirements & Portal Platform

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

NSF
#1541215

4.2 Integrating DrAFTS into the Portal

The *Aristotle AWS Pricing Tool* is operational. This tool helps users compare Aristotle resources to the various AWS alternatives based on performance, cost, and price-performance. Visit https://federatedcloud.org/using/drafts.php to learn more or go directly to the "Aristotle AWS Pricing Tool" at http://169.231.235.92:5000/.

4.3 Integrating Open XDMoD into the Portal

4.3.1 Containerized Application Kernels (AK) on Google Cloud

As a part of a public cloud evaluation, we ran containerized application kernels on Google Cloud. Three different virtual machines (VM) were selected (Table 1): 1) general-purpose CPU with 8 cores, 2) Intel Cascade Lake generation CPU with AVX-512 and with 40 physical cores, and 3) AMD Zen-2 generation CPU with 112 physical cores. The first VM represents a reasonably sized machine for a permanent presence online. The other two represent high compute-capable resources. For the comparison, we use 8 core virtual machines from Aristotle Cloud and several XSEDE HPC systems (Table 2).

Table 1. Used Google Cloud virtual instances configurations. Note that Google Cloud uses hyper-threading (HT) cores for core counting, whereas here we count physical cores

Configuration Name	Physical Cores	CPU Family	Machine Type	Disk Type	Estimated Price, $/ Month
Google-Cloud-8core	8	Intel Haswell+	e2-highcpu-16	pd-balanced	147.45
Google-Cloud-40core	40	Intel Cascade Lake	n2-highcpu-80	pd-balanced	1,678.24
Google-Cloud-112core	112	AMD Zen 2	n2d-highcpu-224	pd-balanced	4,083.91
Google-Cloud-8core-FastStorage	8	Intel Haswell+	e2-highcpu-16	pd-ssd	149.55

Table 2. Google Cloud and Aristotle Cloud VMs, as well as other systems where application kernels were executed.

Resource	Resource Type	Physical Cores	Resource Provider	CPU Family
Google-Cloud-8core	Cloud	8	Google	Intel Haswell+
Google-Cloud-40core	Cloud	40	Google	Intel Cascade Lake
Google-Cloud-112core	Cloud	112	Google	AMD Zen 2
Cornell-redcloud	Cloud	8	Cornel U.	Intel Haswell+

NSF
#1541215

UB-lakeeffect	Cloud	8	UB	Intel Haswell+
UCSB-overcloud	Cloud	8	UCSB	Intel Haswell+
Comet	HPC	24	SDSC	Intel Haswell
bridges	HPC	28	PSC	Intel Haswell
stampede2-skx	HPC	48	TACC	Intel Skylake-X
Bridges-2	HPC	128	PSC	AMD Zen 2
Expanse	HPC	128	SDSC	AMD Zen 2

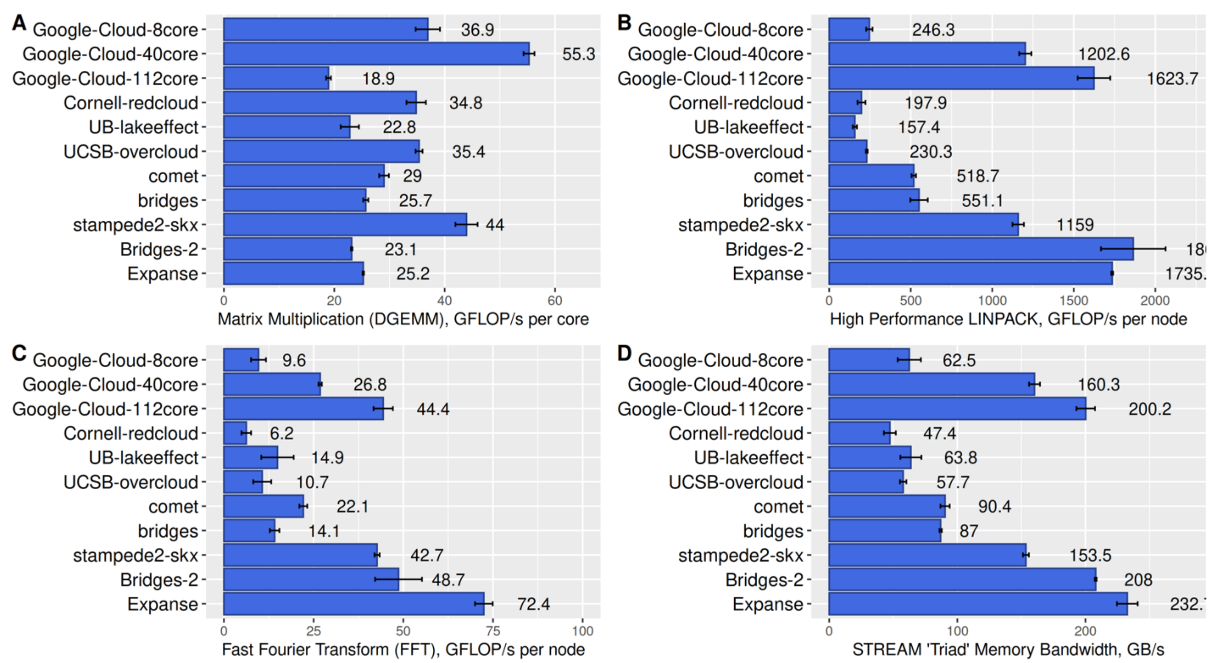


Figure 1. HPCC Benchmark. A. Matrix-Matrix Multiplication. B. LINPACK. C. FFT performance. D. Memory bandwidth.

HPCC is a benchmark suite combining several well-known tests under one package. The selected metrics are shown in Figure 1. The performance of 8-core VMs in Google Cloud and Aristotle Cloud are very similar, with Google Cloud slightly faster. For more compute powerful VMs, the performance is comparable to their counterparts in XSEDE systems. For example, Google Cloud 40 cores VM shows the highest matrix-matrix multiplication per core due to wider SIMD instructions (AVX512) available on that system. TACC Stampede 2 SKX system has a 20 % slower matrix-matrix multiplication performance; this system is also AVX512 capable but is based on older CPU generation and probably has a slower base clock.

In the per-core performance, Google Cloud 112 cores VM is very similar to Bridges-2 and Expanse. These three systems utilize the same generation CPU (AMD Zen 2) and are only different by core counts and base clocks (Bridges-2 and Expanse have the same CPU model). Per-node performance is smaller on Google Cloud 112 cores VM due to smaller core count. HPCG shows similar outcomes as HPCC (Figure 2). Importantly the performance variability for Google Cloud is comparable to Aristotle Cloud and XSEDE HPC resources.

NSF
#1541215

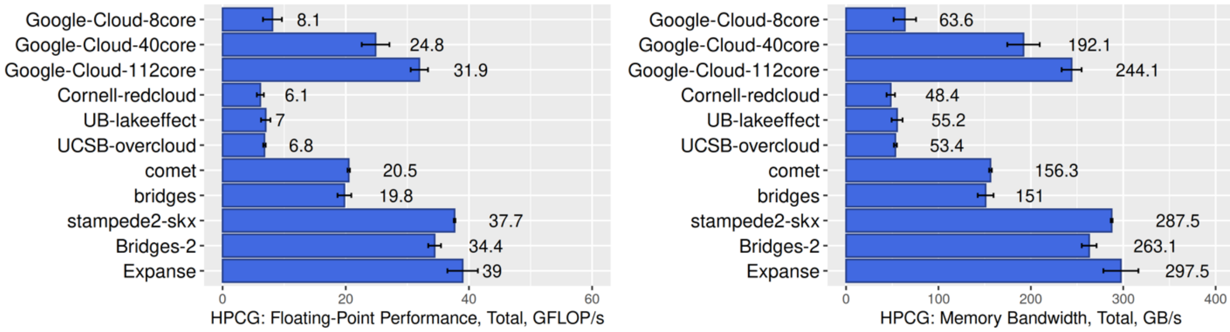


Figure 2. HPCG. Note that on TACC-Stampede-2 SKX an Intel optimized version of HPCG were used.

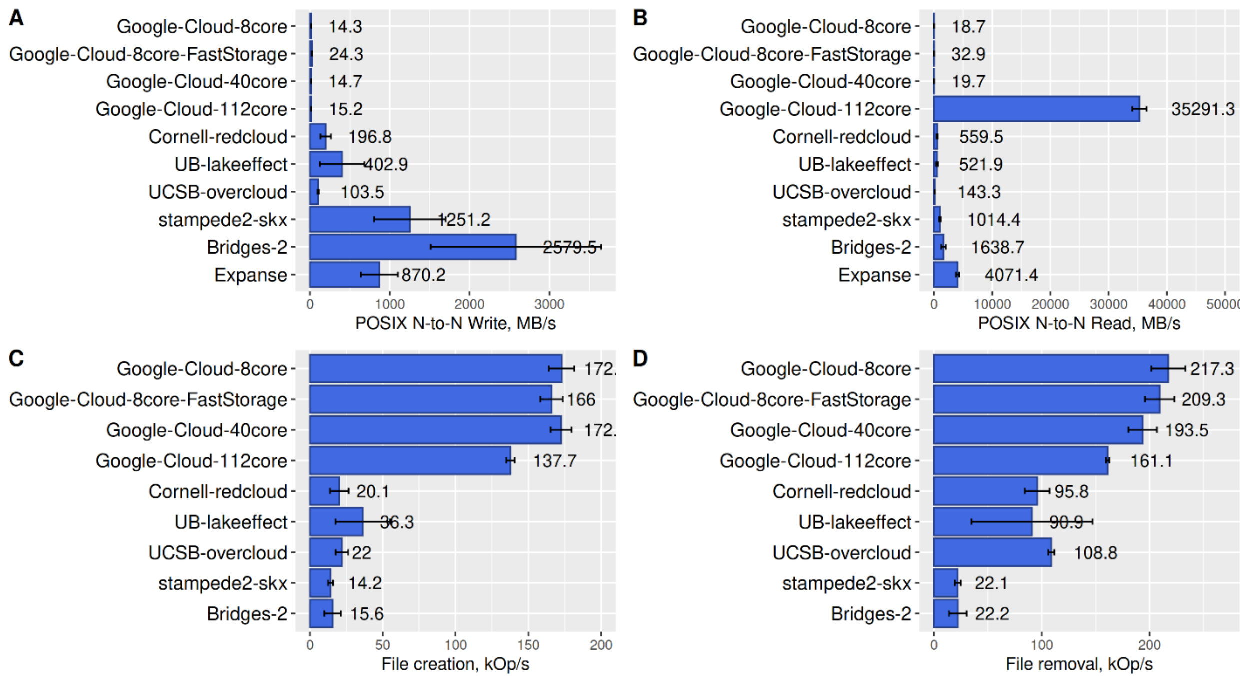


Figure 3. Filesystem IO benchmarking by IOR, measuring bandwidth (A and B) and MDTest, measuring metadata operations (C and D). Measuring filesystem IO is difficult due to disk caching, which hides the low performance of underlying hardware and overestimates the performance measurements. There are several ways to lower the effects of caching. On the HPC system, file writing and reading occurs on a different node. Because our Cloud benchmarks use only one VM instance, such a technique cannot be used. So, the disk cache was dropped within VM (require root privileges), but it is unclear how it affects caching on an actual physical machine.

The Google Cloud filesystem IO bandwidth is moderate for both balanced and SSD persistent storage. The latter doubles the performance, but it is still lower than Aristotle Cloud values, or XSEDE HPC resources. The metadata operations, on the other hand, are significantly better on Google Cloud.

#1541215

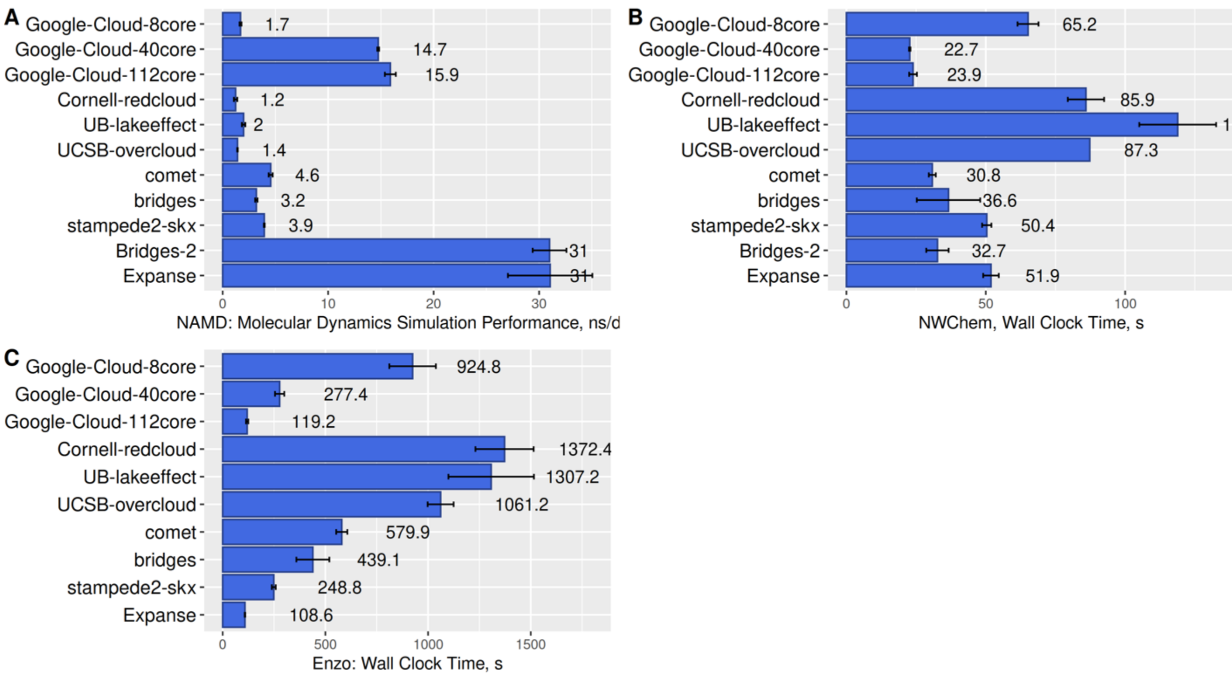


Figure 4. The performance in scientific applications. A. simulation speed in NAMD (higher value is better). B NWChem execution wall time (smaller is better) and C. Enzo execution wall time (smaller is better).

The results from pure benchmarks translate well to the performance in real applications. The performance of Google Cloud in real applications is well in the range of other platforms, especially if one compares platforms with similar computational capabilities. The 8-core machine is almost always better than a similar configuration from Aristotle Cloud. Newer and more compute-capable VMs (40 and 112 cores) have better performance than comparable systems (Bridges-2 and Expanse) in NWChem but are slower in NAMD and Enzo. In all cases, the performance is similar in size and variability.

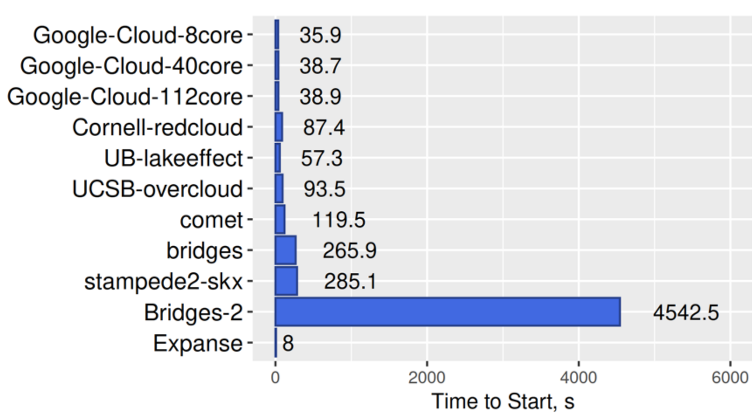


Figure 5. Time to get access to computing resources.

Another important aspect is the waiting time until a resource is available for computation. On the Cloud platforms, it was measured as a time from the instance creation request till the first log-in to the system. On HPC resources, it was measured as time between batch job submission and resource allocation. It is important to note that in the Cloud computing, resources are allocated indefinitely while on HPC resources for a specified time. Because app kernels jobs are very short and use only a single node for this test, their waiting time was short, especially on the new less occupied resource from SDSC (Expanse). Here, Google Cloud has a smaller spin-up time than our OpenStack instances and provides access to compute resources in less than a one-minute timeframe. This is significantly smaller than that on busy HPC resources like PSC-Bridges-2 and slightly longer than on underutilized resources.

In summary, Google Cloud offers a highly competitive alternative for single node computation to HPC. The performance and its variability are within the range of traditional platforms. This report is available at the Aristotle portal.
https://federatedcloud.org/papers/ContainerizedAppsOnGoogleCloud.pdf

4.3.2 XDMoD Cloud Integration

All 3 sites are running Federated Open XDMoD 9.0.
.

4.4 Allocations & Accounting

The federation's database schema is available to the broader cyberinfrastructure community.
https://federatedcloud.org/using/buildyourown.php

5.0 Research Team Support

5.1 Science Use Case Team Updates

Use Case 1: A Cloud-Based Framework for Visualization & Analysis of Big Geospatial Data

Aristotle continues to host the OUTSTEPS integrated community platform and the webGlobe analytical ecosystem. OUTSTEPS is a multi-institution regional research network focused on sustainability of the Lower Great Lakes. A paper describing the digital platform and its capabilities is underway and a research proposal was submitted to the NSF.
https://outsteps.org/

Use Case 2: Global Market Efficiency Impact

Dominik Roesch and his University of Utah collaborators used Aristotle and the financial framework to investigate whether human traders still matter at a time in which trading is dominated by computers. The results were presented at the *Conference for Finance Market Regulation* and *European Finance Annual Meeting*.
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3609007

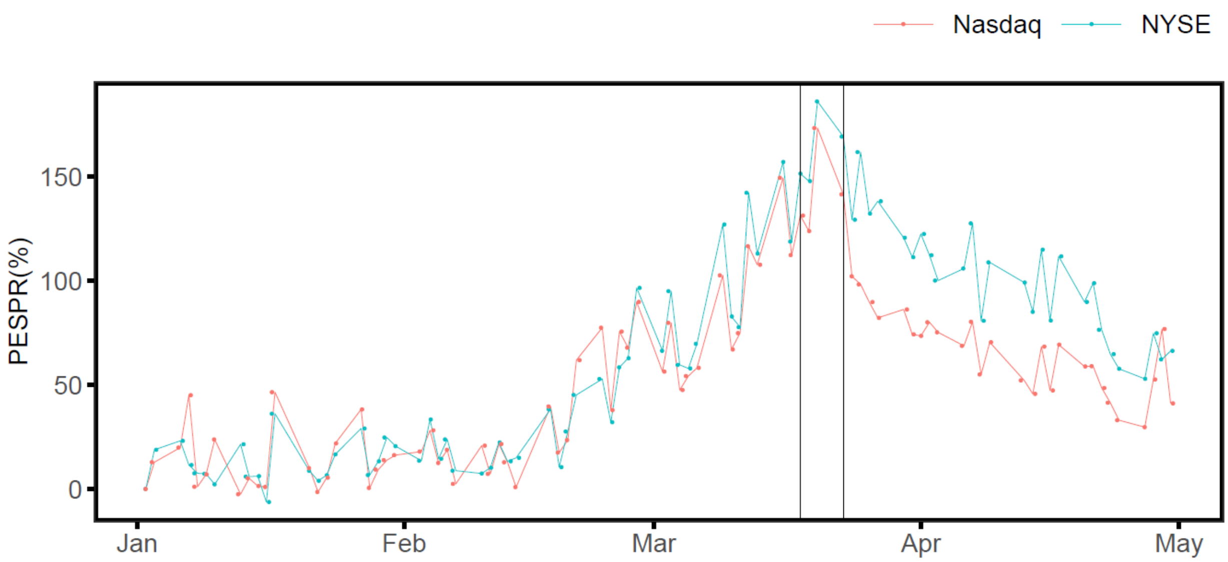
Progress also continued on a LASSO regression predicting short-term stock returns using the whole cross-section of international stocks and an investigation of Cum-Ex trading (labeled as the "Biggest Tax Heist Ever" by the *NYTimes*).

Each of these projects require large datasets and computational power.

NSF
#1541215

"Investor short-termism and real investment" was published in the *Journal of Finance.* https://www.sciencedirect.com/science/article/pii/S1386418121000276

The figure below is from a recent working paper titled "Does the Trading Floor Matter?" which investigates the importance of the New York Stock Exchange Floor and, with it, the importance of people vs. algos (AI).



Daily transaction costs for NYSE listed relative to NASDAQ listed stocks: This figure shows proportional transaction costs in percent for all matched NYSE and NASDAQ listed stocks in our sample. We calculate changes in transaction costs since January 1, 2020 per stock-day and plot daily market cap-weighted transaction costs across all stocks in our sample. Transaction costs are estimated as the trade price across all exchanges in excess of the prevailing midpoint price using national best bid and ask prices. The first vertical line (2020-03-18) indicates the announcement of the event, the closing of the NYSE floor. The second vertical line (2020-03-23) indicates the event day.

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

PI: Sara C. Pryor, Cornell

New WRF Simulations:
Currently our effort is focused on attempting the implementation of a multi-node WRF simulation capability. Partial success has been achieved.

Analysis of Prior Simulations:
Most of our effort continues to be focused on analysis of our existing simulations centered on two key themes:

1) *Use of machine learning approaches for wind gust detection and quantification.* We have compared our new deep learning-based approach with a straightforward (physics-based) post-processing parameterization of wind gusts and statistical approaches that employ linear methods. Our analyses show ANNs exhibit higher skill than logistic and linear regression models for occurrence and magnitude, respectively. Importantly, they exhibit higher skill in forecasting strong (17 to 25.7 ms-1) and damaging (≥ 25.7 ms-1) wind gusts. Deep ANN models capture up to half of all strong and damaging wind gusts at the three airports we have considered: Newark, Boston and Chicago O'Hare. Our work in this arena benefits from the Aristotle architecture in two keys ways; (i) Availability of large RAM. The datasets are multi-dimensional and large. (ii) Availability of multi-processors. Initial network design and testing for the machine learning is being conducted in a hierarchical framework wherein we are seeking to recursively examine the skill of Artificial Neural Networks as a function of (a) number and nature of predictor variables including inclusion of autoregressive terms (from 1 to 16 with all variable combinations considered, i.e. several hundred model forms of variable predictor numbers from; x1&x2 to x1..xn) (b) depth of the network (i.e. number of hidden layers, from 1 to 9, where typically 1-3 is typically regarded as optimal), and (c) the precise data record used for the network training (via bootstrapping, 1000 draws). We are using an informed (or constrained) sparse matrix approach where (c) is only applied to models from (a) and (b) that exhibit statistical skill above a threshold. Initial work largely focused on optimal network settings and so the time of training and testing of the network at a single site on a single processor of 3.1 minutes (on average) has not been too problematic. However, given the amount of uncertainty space we are seeking to explore (n > 15000), running on a single core would mean over a month of processing time per individual airport location.

2) *Analysis of our first set of simulations of wind farm wakes from the east-coastal offshore lease areas using ultra-high resolutions with WRF.* The simulations were performed on NERSC-Cori but were ported onto Aristotle for analysis. Preliminary analysis results were presented at the Wind Energy Science Conference in Hannover, Germany: Barthelmie R.J., Shepherd T.J. and Pryor S.C. (2021); Offshore wakes in the U.E. east coast lease areas. Our work in this arena benefits from the Aristotle architecture in three keys ways; (i) Availability of large RAM. The WRF output is multi-dimensional and large. (ii) Availability of multi-processors for analysis speed using parallel processing. (iii) Availability of large disk volumes so all WRF output can be hosted for analysis.

Journal manuscripts in review:
Coburn J.J. and Pryor S.C.: Do machine learning approaches offer skill improvement for short-term forecasting of wind gust occurrence and magnitude? *Weather and Forecasting*.

Journal publication:
Aird J.A., Barthelmie R.J., Shepherd T.J. and Pryor S.C.: WRF-simulated low-level jets over Iowa: characterization and sensitivity studies. *Wind Energy Science.*
https://wes.copernicus.org/articles/6/1015/2021/

Use Case 4: Transient Detection in Radio Astronomy Search Data

The Singularity container developed by use case support team member Peter Vaillancourt has been used by PhD candidate Akshay Suresh on XSEDE's Bridges and allowed a very simple transition to Bridges-2, where the work is ongoing. Suresh described the transition as "seamless" which is often *not* the case for moving between specialized computational resources. The container consists of radio astronomy software that combines pipeline components developed for pulsar and fast radio burst (FRB) detections.

Use Case 5: Water Resource Management Using OpenMORDM

Bennett Wineholt received a measurement script to run OpenMORDM with step timing to quantify overhead.

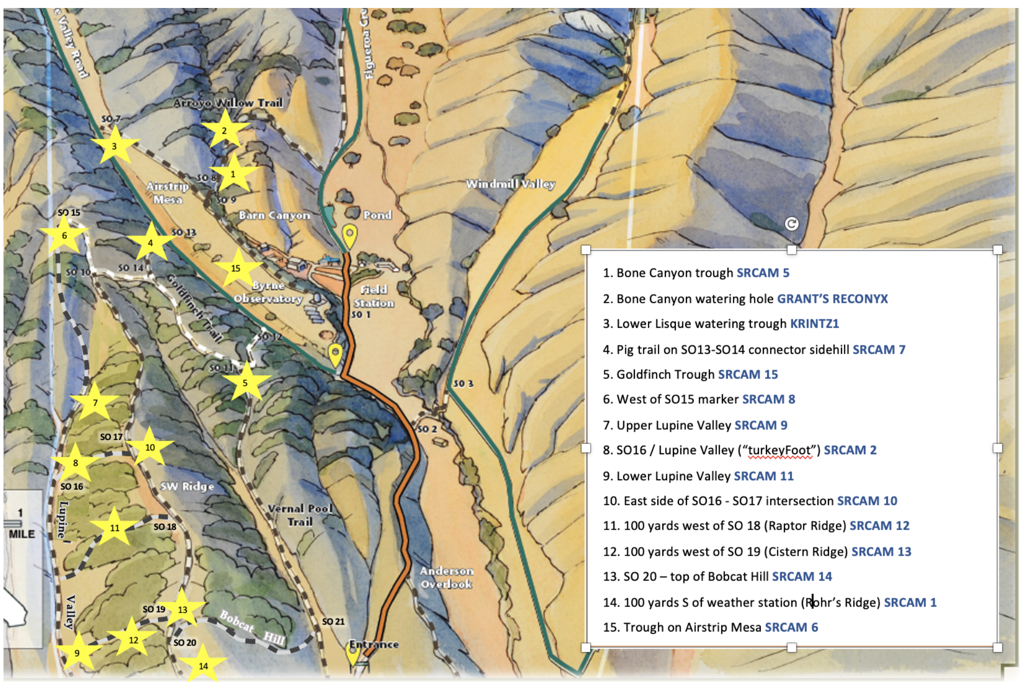
Use Case 6: Mapping Transcriptome Data to Metabolic Models of Gut Microbiota

This research project culminated with the publication: Nana Y.D. Ankrah, Brandon E. Barker, Joan Song, Cindy Wu, John G. McMullen II and Angela Douglas, "The Predicted Metabolic Function of the Gut Microbiota of *Drosophila melangogaster*" in the *mSystems Journal*.
https://journals.asm.org/doi/full/10.1128/mSystems.01369-20

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

Sedgwick Reserve:

Work at the Sedgwick Reserve remains on hold pending the approval of a requested exemption from COVID restrictions that would permit the Science Team to visit the site.



Camera traps are collecting data at multiple Sedgwick locations

"Sparta: A Heat-Budget-based Scheduling Framework on IoT Edge Systems" was accepted by the *International Conference on Edge Computing.*
https://sites.cs.ucsb.edu/~ckrintz/papers/edge21.pdf
UCSB Edible Campus:

The UCSB Edible Campus Food Security project continues with at-home research by the students. Currently, the Science Team is developing an image analysis system that attempts to correlate windsock images with windspeed and direction measurements to provide the Edible Campus with low-cost, high-density airflow measurements.

Citrus Frost Prevention
Lindcove Research and Extension Center, Exeter, CA:

Implementation of a CFD model for airflow through the Citrus Under Protective Screening (CUPS) screenhouse structure is proceeding. Currently, a preliminary model captures air velocity and heat transfer through the porous mesh vertical wall with an open top. Science Team graduate students are now enhancing the model to account for a mesh top and an incident wind wall that is canted 45 degrees (to reflect the true geometry of the structure). Once the geometry is successfully incorporated, model development will turn to capturing vortical effects.

A validation experiment is planned for the basic velocity and heat transfer model for the time period immediately following the lifting of COVID restrictions on university travel. As part of the instrumentation needed to develop the predictive CFD model, the Science Team created a data visualizer for the growers who are collaborating with the team on the project. Hosted in Aristotle, the prototype is undergoing user testing ahead of a production release scheduled for next month.

6.0 Community Outreach and Education

6.1 Community Outreach

- The Aristotle team participated in a virtual tutorial at *PEARC'21* with IU colleagues titled "A Deep Dive into Constructing Containers for Scientific Computing and Gateways." https://github.com/XSEDE/Container_Tutorial/blob/main/Gateways2020/Day2_3_AdvancedTopics.pdf)
  This tutorial was also delivered at the *Science Gateways Community Institute Coding Institute 2021*and at *eScience 2021*. It is scheduled for *SC21* next.

- A July *NYTimes* story explained what scientists know about the beta variant. It includes insights from Laith Abu-Raddad, an infectious disease researcher at Weill Cornell Medicine, whose research was made possible by Aristotle Red Cloud.
  https://www.nytimes.com/2021/07/19/health/beta-variant-covid-19.html?searchResultPosition=1

- Technologies and techniques developed by the project team are available on the Aristotle portal. We will respond to future inquiries and share our knowledge and experiences as requested. https://federatedcloud.org/using/technologies.php.

6.2 Education

- Cornell launched a *YouTube* site in August 2021 to augment their wiki-based training with how-to videos. The first videos in the ongoing series include how to connect to Linux instances, launching an instance, how to create a security group, how to access the OpenStack web interface, and how to create key pairs.
  https://www.youtube.com/channel/UCVPGMVWhp3sqWZFU5NntjTA

- Former Aristotle REU and Cornell undergrad Jeffrey Lantz ran a containerized application with Intel compilers and MPI over multiple VMs in public clouds. A video of Lantz presenting his findings—"WRF in the Public Cloud—is available.
  https://www.youtube.com/watch?v=32ISd8LUQRw&t=1263s

- Students at the Aristotle sites learned about and, in several cases, created new cloud and edge computing technologies. Opportunities arose for students when use case projects required special capabilities such as automated deployment or IoT tools. For example, UCSB PhD student Wei-Tsung Lin built a serverless computing platform for multi-scale and heterogeneous environments using geo-replicated logs. Lin was awarded his PhD in June.

- During Spring 2021, UB professor Mohammed Zia used the Aristotle cloud service to host JupyterHub with the nbgrader module enabled on an Ubuntu image for his *Programming and Database Fundamentals for Data Scientists (EAS 503)* class. The nbgrader module allows him to create and grade programming assignments in JupyterHub, a streamlined method that enables more frequent distribution of assignments. Aristotle is cost-effective because Zia can choose the instance type based on the expected class size. There are usually 50 students in the spring semester and 120 students in the fall semester. Zia selects the instance type he needs based on the class size, and after the semester is over, deletes the instance. Ease of deployment and the ability to handle all of the student's interactive Python needs in one place are very beneficial.

- Aristotle science use case projects provided students with unique opportunities to develop their skills and apply those skills to their own research. For example, use case scientist Dominik Roesch taught three University at Buffalo PhD students to use the OneTick Time-Series financial framework with data hosted on Aristotle. The students are now using those skills to conduct PhD-level research.

- Aristotle how-to user guides were developed on GitHub and are publicly available on the Aristotle portal. A "Getting Started with Kubernetes" addition is planned.
  https://federatedcloud.org/using/gettingstarted.php
  Information on how to create, access, and set up Linux and Windows instances; set up networks and share files; and use the OpenStack Command-Line Interface was added.

- Aristotle documentation on how to build your own federation is available and includes our database schema, Globus user credential lookup sample code, Single Sign-On with Globus Auth, and hardware and software recommendations.
  https://federatedcloud.org/using/buildyourown.php

7.0 Future Plans

The “New York Region” of the Jetstream 2 cloud will be installed at Cornell by November and will eventually be integrated with Aristotle Red Cloud. Cornell will use this capability to further its exploration of federated clouds and to make OpenStack enhancements. Lessons learned will be disseminated to the Jetstream 2 partners and the broader community. Cornell was included in the Jetstream 2 project as a direct result of its Aristotle Cloud Federation work. Cornell’s Red Cloud will support future research in areas such as COVID-19 epidemiology, HIV modeling, cryptography, robot perception, and the understanding of the impurity of atoms in semiconductors and the ease with which ions can move in and out of batteries and other energy devices.

The University at Buffalo’s OpenStack cloud will continue to provide essential research support for UB faculty and their national collaborators. Support for containers and Kubernetes orchestration were added to prepare for new research workloads in the cloud. Multiple NSF grant proposal submissions were submitted that propose to utilize the work and infrastructure that has been put in place with the Aristotle project, thus requiring less hardware infrastructure funds and no start up time for building out the CI platform. UB’s NSF award, “Collaborative Research: CCRI: New: Medium: A Development and Experimental Environment for Privacy-preserving and Secure (DEEPSECURE) Machine Learning,” will leverage the federation capabilities of UB’s cloud to branch out to Old Dominion's OpenStack cloud, developing a Software Development Environment and Multi-User Experimental Chamber for securing ML/AI Research.

The Aristotle project germinated UCSB’s cloud infrastructure and spurred cloud computing forward in computer science, engineering, and the arts and sciences. Use cases covered a broad range of research and instruction such as the application of IoT, edge, and cloud for digital agriculture, the use of remote data to study kelp forest and atmospheric lifecycle dynamics; research support for Media Neuroscience and R-Shief; enablement of the GauchoSpace Learning Management System, and sandbox environments for capstone class projects in Computer Science, Data Science, and Statistics and Applied Probability. UCSB will apply the techniques and technologies developed for Aristotle use cases to support future data science and ML projects; build upon current resource capacity, computing cores, and storage; and, offer new data, message queuing, and Kubernetes services to the wider community.