Newsletter

Spring 2017

Center for HIGH PERFORMANCE COMPUTING
THE UNIVERSITY OF UTAH

Research Highlight

Efficient Storage and Data Mining of Atmospheric Model Output Using the CHPC Pando Archive

by Brian Blaylock and John Horel, Department of Atmospheric Sciences

Brian Blaylock

Graduate student, Department of Atmospheric Sciences

John Horel, Ph.D.

Professor, Department of Atmospheric Sciences

The University of Utah

CHPC recently added 1PB of storage as part of its pando archive storage system at the University of Utah's Downtown Data Center. This archive system is similar to Amazon's Simple Storage Service (S3) and provides a web service interface to store and retrieve data objects. Our group in the Department of Atmospheric Science purchased 30TB in CHPC's pando system to test its suitability for several research projects. We have relied extensively over the years on other CHPC storage media such as the tape archive system and currently have over 100TB of network file system disk storage. However, the pando system is beginning to meet several of our interwoven needs that are less practical using other data archival approaches: (1) efficient expandable storage for thousands of large data files; (2) data analysis using fast retrieval of user selectable byte-ranges within those data files; and (3) the ability to have the data accessible to the atmospheric science research community.

Long-term archives of operational numerical weather prediction forecasts and weather and climate research simulations require vast amounts of storage. Atmospheric scientists rely heavily on national data centers (e.g., National Centers for Environmental Information, NCEI) to access retrospectively operational weather forecast model output generated by the National Centers for Environmental Prediction (NCEP) of the National Weather Service. Managing local archives of such model output can quickly exceed what university researchers can afford to maintain in university computational facilities. Further, while simply warehousing data is common in our field using many different types of storage media, efficiently analyzing large volumes of archived model output has generally been very difficult to do. In addition, making such local archives accessible to research groups outside the host university is often impractical to allow.

Beginning April 2015, we began downloading output from NCEP's High Resolution Rapid Refresh (HRRR) model to initialize research simulations for an air quality study that summer (Blaylock et al. 2017, Horel et al. 2016). The HRRR is the highest spatial and temporal resolution forecast system run operationally in the United States and is well suited for diverse research applications of interest to our group. It provides two-dimensional meteorological fields (temperature, wind, moisture, etc.) at 3 km horizontal resolution over the continental United States (1.9 million grid points) every hour (the current hour is referred to as the analysis) and hourly forecasts extending out 18 hours. The analysis and forecast grids are archived in a highly efficient "grib2" binary format commonly used in our field.

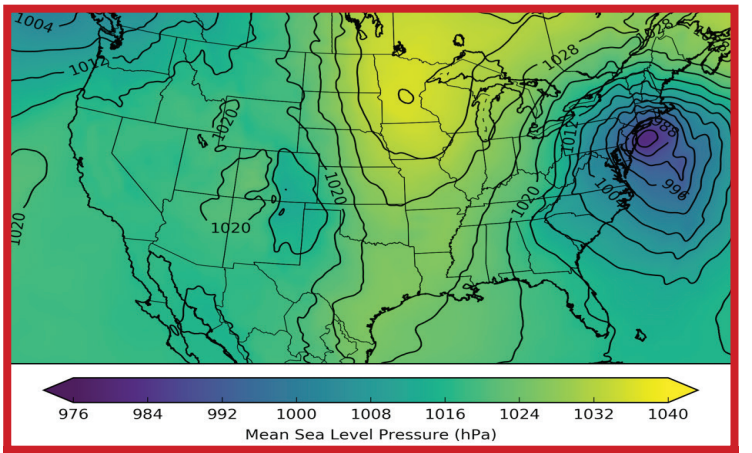


Figure 1.
Sea level pressure analysis from the operational HRRR at 1 PM March 14, 2017 with unusually low pressure associated with a major New England snowstorm.

As a simple illustration, Figure 1 shows the sea level pressure across the HRRR domain at the time of a major New England snowstorm.

We are currently using roughly 20TB in the pando system, which includes an archive of the operational HRRR model analyses for two years. The forecast hours have been added to the archive since summer 2017. We also recently began archiving additional analyses from the experimental HRRR model as well analyses and forecasts from the HRRR-Alaska version of the model. At our current rate of adding roughly

65GB of HRRR model data per day, we will exceed our current pando allocation by mid-August 2017! All of the files in the HRRR archive are accessible to researchers at the University of Utah and off campus interactively through a web page or the user can create automated download procedures using wget or curl. Since individual files are large (about 120MB for two-dimensional fields, or about 300MB for three-dimensional fields), and the file contents may contain more variables than the user needs, users can download only the variable they need if the byte range is known. (See *http://hrrr.chpc.utah.edu/*).

As with Amazon S3 objects, objects stored in the CHPC pando system must be retrieved from the archive in order to be used, i.e., it is not possible to manipulate the objects directly from CHPC workstations and servers as data files are stored locally on the network file system. We have developed python multi-processor procedures that rely on basic curl commands to efficiently access the HRRR files from a single dedicated CHPC server. Initial testing was done in order to obtain basic statistics (minimum, mean, maximum, and percentiles) of meteorological variables (temperature, wind speed, snow cover, lightning, etc.) within the 2-year archive of HRRR analysis grids. Such statistics will be used to help identify erroneous environmental observations that we collect from over 30,000 locations and make available for diverse applications (*https://mesowest.org*). For example, computing the minimum, mean, and maximum wind speed from nearly 17,000 hourly analyses at the 1.9 million grid points in the operational HRRR model was done in less than 15 minutes using 30 processors. Computing multiple percentile values, such as the 95th percentile for wind gusts, separately for each hour of the day as shown in Figure 2 required roughly 50 minutes. As the HRRR archive continues to grow, new procedures will need to be developed to effectively develop such statistics within the memory constraints of typical computational servers.

The CHPC pando storage archive has made it possible for us to efficiently archive, access, and analyze a large volume of atmospheric model output. Several researchers outside the University of Utah have already discovered its utility in the short time that the archive has been available. Our research group uses Amazon AWS including its S3 archive for other applications that require uninterruptible computational resources. For our research applications that can tolerate occasional down time, the pando system is more cost effective and provides faster access to our data archive.

Since nearly all agencies now require data management plans and long-term storage of research results, the pando archive may help to effectively meet those requirements. Of particular benefit would be for the University to provide support to CHPC to become an official data repository, which is required for publishing descriptions of archived data sets in data journals (e.g., Jacques et al. 2016). *This research is supported by NSF Grant 1443046.*

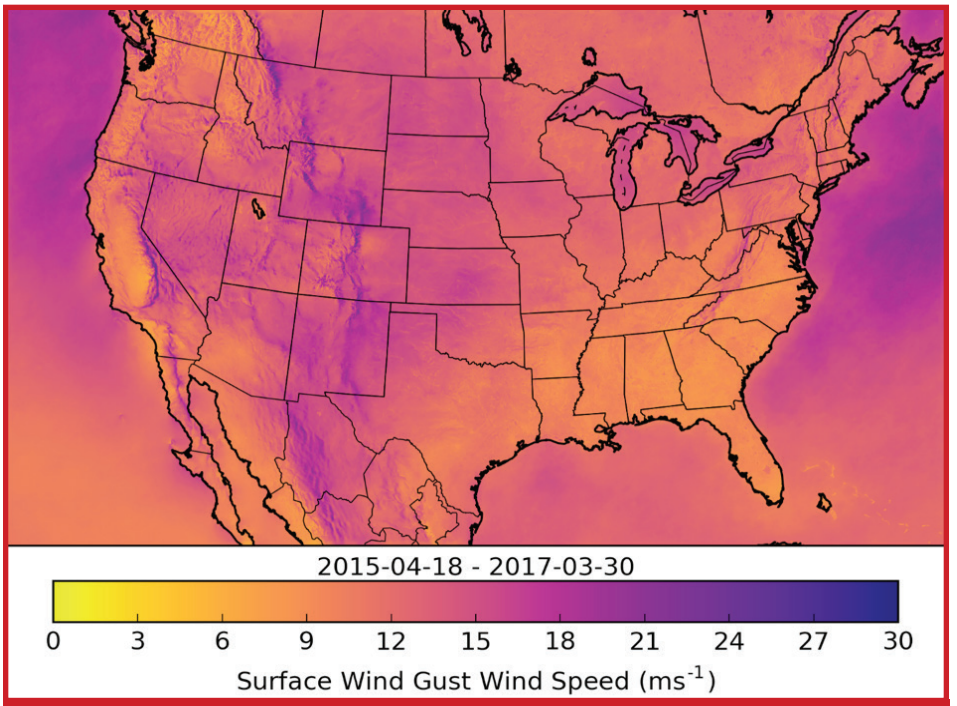


Figure 2.
95th percentile of the 10 m surface wind gusts analyzed by the operational HRRR at 23 UTC (4 PM MST/5 PM MDT) during all days between April 18, 2015 and March 30, 2017.

Blaylock, B., J. Horel, E. Crosman, 2017: Impact of Lake Breezes on Summer Ozone Concentrations in the Salt Lake Valley. J. Appl. Meteor. Clim., 56: 353-370. doi: 10.1175/JAMC-D-16-0216.1

Horel, J., E. Crosman, A. Jacques, B. Blaylock, S. Arens, A. Long, J. Sohl, R. Martin, 2016: Influence of the Great Salt Lake on summer air quality over nearby urban areas. Atmospheric Science Letters. 17, 480-486. doi: 10.1002/asl.680

Jacques, A., J. Horel, E. Crosman, F. Vernon, J. Tytell, 2016: The Earthscope US Transportable Array 1 Hz Surface Pressure Dataset. Geoscience Data Journal, 3: 29–36. doi: 10.1002/gdj3.37

For more information on the newly available pando archive storage system, please see next article.

Technical Highlight: Archive Storage Solution at CHPC

by Sam Liston

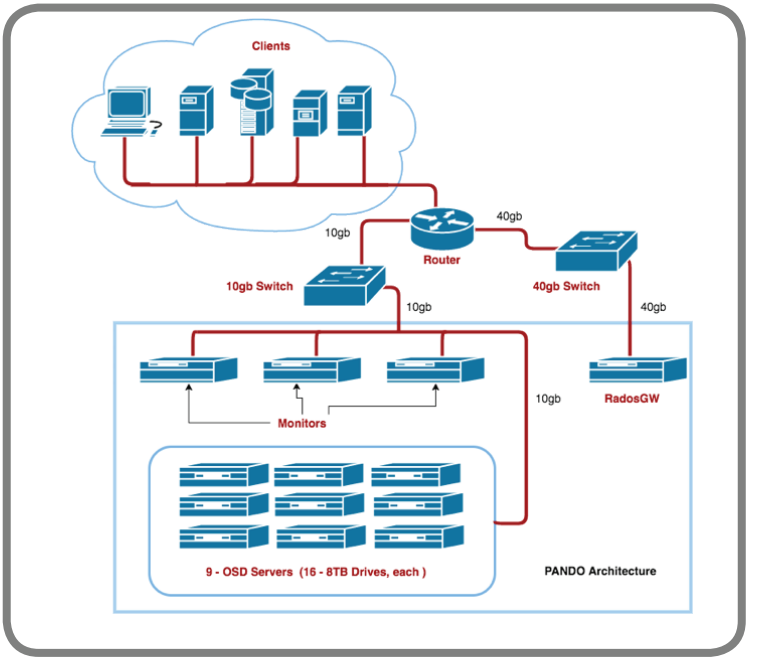
As mentioned in the previous article, CHPC recently added a PB of archive storage. This was partially in response to the growing demand for storage, particularly group space. As a result of this rapid growth, the backup capacities at CHPC have not been able to scale at a proportional rate, leaving a good portion of the total data without backup. This archive storage, named pando, an object-based storage system, provides a place for researchers to store a secondary copy of their data. At 1.02PB in usable capacity, pando comes at cost lower than our group storage offering, has greater resiliency characteristics than our other storage systems, and is much more accessible than a traditional backup. In this solution, researchers are responsible for moving data in and out of the archive, removing some administrative burden.

While planning this solution, we wanted it to have the ability to scale to an immense size. In order to do that we had to find a solution that addresses the fundamental scaling issues that both traditional file systems and RAID (Redundant Array of Independent Disks) sets have. In a standard RAID design as the drive size increases the time for the array to rebuild after a failure also increases. Similarly, as maximum file system capacities increase, the time for recovering and repairing a file system in the event of an error or corruption also increases, to the point where it may require many days to find and repair the errors. An alternative method to manage and organize files at massive scales is therefore necessary.

Object storage addresses these design shortcomings. Unlike a traditional file system where the efficient organization of files and directories falls solely on the user, an object system puts a layer of software abstraction between the user and the underlying file systems. This system organizes files and directories as objects, in a formulated, efficient, flat structure. The object-based software layer used on our system is called Ceph, an open-source solution initially developed at the University of California Santa Cruz and is now supported by an active development community and RedHat. Ceph continues to be developed by a strong community, providing new releases, bug fixes and support.

The beauty of Ceph is in its design. Resiliency is configured at the object layer versus the system layer. Every object is replicated or made redundant according to configuration choices. This can be set up as N replicated copies or a particular factor of erasure coding defined as K+M, in which an object is broken up into K data chunks plus M resiliency chunks. When the Ceph cluster is configured, pools are first created to provide structure to the storage capacity according to the replication and/or redundancy schemes chosen.

In the CHPC Ceph installation there are nine Object Storage Device (OSD) nodes which house the actual data, as shown in the figure below, allowing for 6+3 erasure coding. In this configuration three servers worth of disks can fail before data is lost. In addition, there are three monitor nodes that keep and maintain the map of the objects in the system, a gateway node that offers a S3 interface to users, and an administration node.



Within the pool configuration there is the idea of placement groups. A placement group defines the placement of objects across the system in a manner that ensures a loss of some number of drives (based on the replication used) does not result in any loss of data. Because of this an individual file system becomes disposable. If an individual file system becomes corrupt, rather than check and repair the file system, it is faster and easier to logically remove that drive from the system, reformat, and add it back in. In this process Ceph recognizes that the objects on that drive are missing, that some placement groups are not fully resilient, and then immediately begins redistributing the objects in order to restore the necessary redundancy.

Ceph allows for easy expansion of the backend object storage with the addition of more OSD nodes. This allows for seamless, transparent migration of data as old hardware is vacated or retired and new hardware is added and populated. In most traditional storage systems, data must be manually migrated from one generation of hardware to the next. This feature was key in developing an archive that could be in production through a period of time that

extends beyond the various vendor warranty lengths and through several generations of hardware.

As an archive solution pando was created to allow users to manage the movement of data in and out of the system, and is therefore accessible via an S3 API. This API can be accessed directly or referenced in a user written application. In addition, there are three recommended tools for moving data in and out of the archive space: **rclone** (*https://www.chpc.utah.edu/documentation/software/rclone.php*) which can be thought of as rsync for cloud platforms, **Globus** (*https://www.chpc.utah.edu/documentation/software/globus.php*) via an endpoint designed to work with Ceph, or **s3cmd** (*http://s3tools.org/s3cmd*) a command-line tool for interacting with s3 storage. Users can also program as part of their workflow directly with the S3 interface as well as use tools to change permissions on objects in the system to make them "public" and web accessible via a URL. The archive space is not mounted on CHPC resources as the group space option is, thereby requiring the user to move the data from the archive before it can be used.

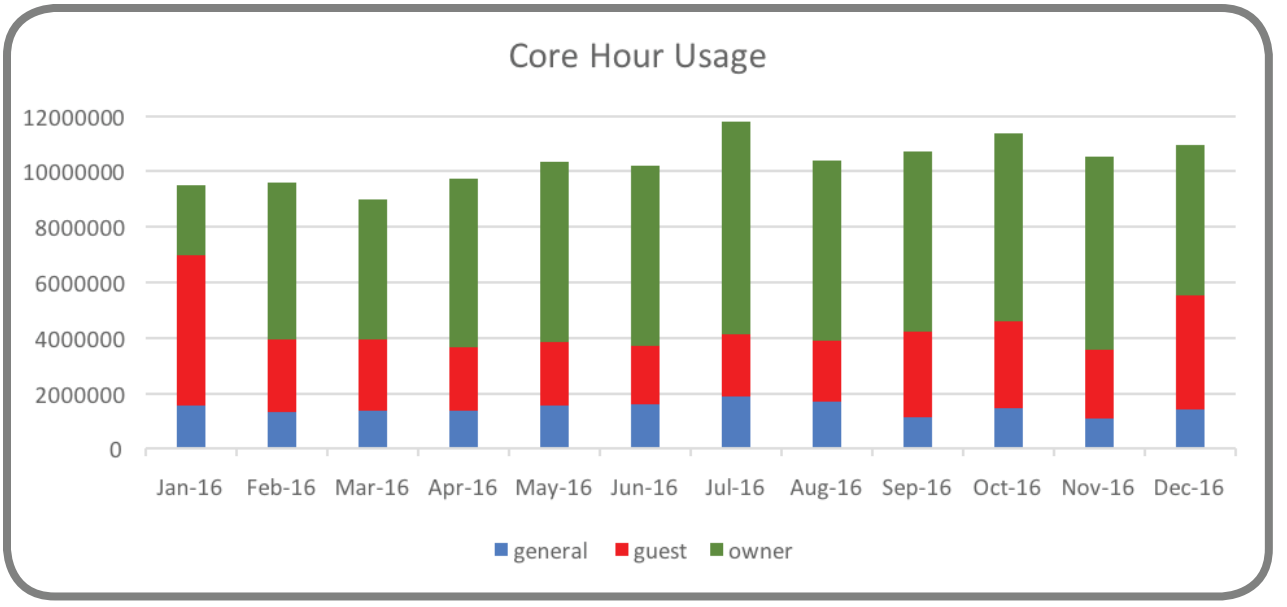
A more detailed write up on the technical aspects of CHPC Ceph offering is available on the CHPC website at *https://www.chpc.utah.edu/documentation/white_papers/*.

Space can be purchased on pando for $120/TB for 5 years. If you are interested in purchasing space, or would like to either test or learn more about pando, please contact ***issues@chpc.utah.edu***.

CHPC Growth Statistics

There has been a tremendous growth in CHPC resources and their usage to support research computing at both the University of Utah and Utah State University.

- *99 new owner compute nodes and 9 new owner interactive nodes added to kingspeak in 2016*
- *Added 85 new research groups and nearly 550 new user accounts in 2016*
- *Increase in number of groups owning nodes as well as number of groups with general allocation*
- *Nearly 50% increase in core hours provided from 2015 to 2016 – from 84 M to 124 M*



CHPC Presentations

The Summer 2017 CHPC Presentation Schedule is now available.

For details see
https://www.chpc.utah.edu/presentations/Summer2017CHPCPresentationSchedule.php
No registration is required except for the XSEDE Summer Bootcamp.

Technical Highlight

Virtualization Containers for Application Deployment and Research Reproducibility

by Martin Čuma

Research computing environments can be complex. Many programs have intricate dependencies, old software may not be supported on newer operation systems (OS), and researchers may need to package the data together with their programs. None of these cases map well to the traditional static hardware-OS environment common in research computing resources, specifically high performance computing (HPC) systems. Encapsulating the application and/or the user environment through OS virtualization offers ways to address such situations.

Operation system virtualization vs. virtual machines

With the emergence of multi-core CPUs a decade ago, many system administrators moved to running multiple virtual machines (VMs) on a single hardware server (Figure 1). While the hardware level virtualization approach is useful for running multiple independent servers, such as for basic IT infrastructure, VMs are unnecessarily heavy for isolating applications or user environments.

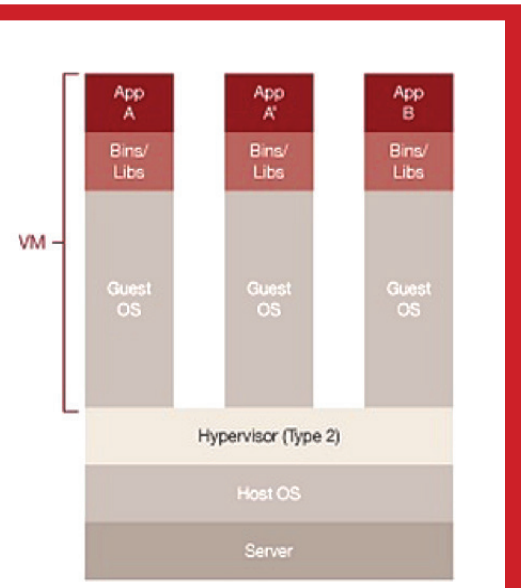


Figure 1.
Schematic of hardware virtualization

Lighter weight operation system level virtualization allows for the running of multiple isolated OS instances (guests), under a server OS (host), as depicted in Figure 2. The guest OSes, also called containers, are isolated from each other, but, share the base host OS and, when appropriate, other parts of the host OS, such as binaries, libraries, etc.

There are numerous container approaches, the most popular being Docker (*https://www.docker.com/*). However, Docker has several design features that do not work well in HPC environments. A recent alternative container approach, Singularity (*http://singularity.lbl.gov/*), addresses these concerns and has gained significant traction in the HPC community. Singularity is being actively developed with the support from Lawrence Berkeley National Laboratory. Singularity integrates well with existing HPC resources, allowing programs in the container to use the same file systems (home, scratch); it also has support for HPC hardware (InfiniBand, GPU) and software (job schedulers like SLURM, MPI). Thanks to the tight coupling to the host OS, the performance overhead of the containers is minimal, if any. Containers are portable and sharable; they encapsulate both the OS and underlying data in a single file that can be built on one OS and run on a resource that uses another. One thing to keep in mind is that Singularity currently only supports Linux, so, we cannot create Windows or MacOS based containers and run them on our Linux based HPC resources.

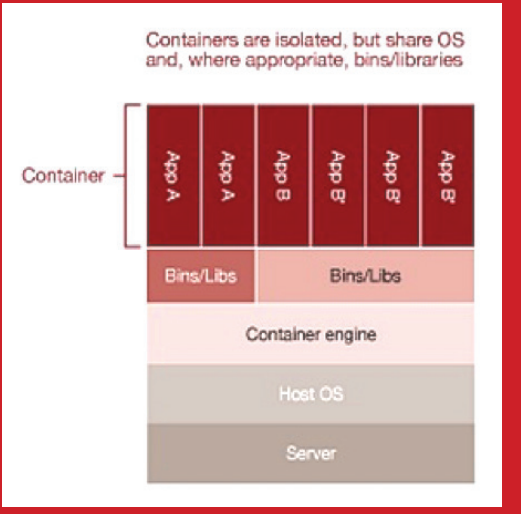


Figure 2 .
Schematic of OS virtualization

The benefits that containers provide include:

1. ***Easier application installation***: When the installation is complex, e.g., there are many dependencies, a container can be built using the OS for which the binaries exist. Similarly, a container can be used for a program that is dependent on a specific type and version of an OS, and/or software stack which is not available in the default HPC environment.
2. ***Portable software development***: Code can be developed on a local computer using whichever Linux distribution is preferred, and then packaged in a container to be run in the HPC environment.
3. ***Reproducibility***: The compute environment, the application and the data can be packaged in a container so that others can reproduce the result.

As the complexity of research applications increases, containers will become an indispensable tool in the management of the software stacks. We envision a rapid adoption of containers both for programs that CHPC staff build and for programs deployed by our users.

Containers at CHPC

A CHPC user has two options when it comes to Singularity containers.

The first option is that a user can use a container that CHPC built. We provide containers for hard to build applications and for complex software pipelines, especially those that use and mix interpreted languages such as Python and R. Several application installations making use of Singularity containers are Tensorflow GPU, SEQLinkage, and bioBakery. Moving forward, we see the number of containers that we provide growing.

The second option is for the user to build the container. To do this the user needs to have administrative privileges. Therefore, users cannot build a container on CHPC administered Linux machines, but rather must use their own computer, on which they have administrative rights. Furthermore, if one does not run Linux on their personal computer, it needs to be installed, ideally in a VM. We describe a relatively straightforward way to set up a Linux VM on Windows and MacOS, and install Singularity at *https://www.chpc.utah.edu/documentation/software/containers-localbuild.php*.

The container build and use process is depicted in Figure 3. When building, one first needs to create the empty container with **sudo singularity create**, and then bootstrap it (install the OS and necessary programs) with **sudo singularity import** or **sudo singularity bootstrap**. For additional details on this process see our main container help page, *https://www.chpc.utah.edu/documentation/software/containers.php*. If applicable, we recommend to base any local container builds on definitions provided at our github page, *https://github.com/CHPC-UofU*, or elsewhere.

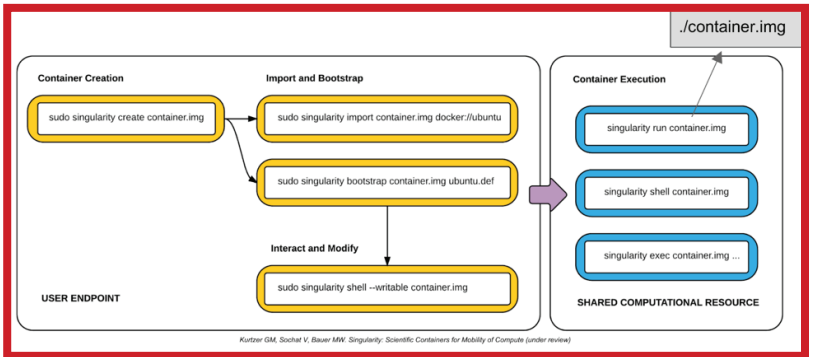


Figure 3.
Singularity workflow

Once the Singularity container is built, one can either open a shell in it with **singularity shell**, run a program from the container with **singularity exec**, or run the predefined default container program with **singularity exec**, or simply execute the container file. Singularity developers have also recently launched a hub for singularity images, *https://singularity-hub.org/*.

As mentioned above, we are planning to package complex software environments into Singularity containers in the near future and we encourage our users to do the same. We would love to hear from you if your research could benefit from a use of containers, or about your experiences with Singularity. Please contact us via *issues@ chpc.utah.edu*.

CHPC Awarded Two Grants

NIH S10 OD-021644: "From genomics to natural language processing: A Protected Environment for Research Computing in the Health Sciences"
(4/15/17-4/14/18)

This NIH S10 shared instrumentation grant, (*https://dpcpsi.nih.gov/orip/diic/shared_instrumentation*) will fund a replacement for the current protected environment. This new protected environment will provide research computing and data management capabilities for researchers to properly manage, secure, and analyze HIPAA regulated protected health information and other sensitive or restricted data.

NSF ACI-1659425: "CC* Cyber Team: Creating a Community of Regional Data and Workflow Cyberinfrastructure Facilitators"
(4/1/17-3/30/20)

The second grant, funded by NSF, is a joint effort between The University of Utah, University of Colorado at Boulder and Colorado State University – all member institutions of the Rocky Mountain Advanced Computing Consortium (RMACC). This grant will fund a "CyberTeam" of data and workflow facilitators, one at each of the three institutions, to provide support for RMACC by assisting them with data and workflow reuse and management. The facilitators will be chosen to have complementary skills and expertise, and will focus on data curation and metadata, and data and compute workflows, including protected information.

User Services

Open Science Grid Application- Prediction of Mass Spectra of Entries in Natural Product Database

by Anthony Mills, Dale Forrister, Gordon Younkin, and Tom Kursar ; Department of Biology

Plants are not defenseless organisms as they are sometimes perceived to be. In fact, they defend themselves with a wide array of mechanisms including the synthesis of toxic chemical compounds. Our lab studies how defensive traits influence the evolution of Inga—a genus of tropical tree in the legume (Fabaceae) family.

Presently, the principal limitation in our research is identifying the secondary metabolites synthesized by Inga. Because we have thousands of compounds in our samples, the traditional methods of purification by HPLC and structure elucidation by NMR for every compound would be prohibitively expensive and extremely time-intensive. Tandem mass spectrometry (MS/MS) which produces a unique fragmentation fingerprint for every molecule in a sample represents an ideal tool for compound identification on this kind of scale. However, this method would require a database of MS/MS spectra generated from known natural products, and although a number of natural products databases do exist, none of them contain MS/MS data.

The UNPD (Universal Natural Products Database) contains ~230,000 compounds which could—in principle—be downloaded and fragmented in-silico to produce the needed MS/MS spectra by using the Competitive Fragment Modeling program CFM-Predict which is capable of predicting MS/MS spectra for a list of input molecules. These predictions are computationally intensive and failed to run on our standard lab computers. We then turned to CHPC. Working with Wim Cardoen, CHPC Scientific Consultant, we did a number of tests on Ember. However due to the limited number of nodes available at a given time, Wim realized that this project could be more efficiently completed in a high-throughput (HT) rather than high-performance (HP) compute environment.

Wim introduced us to the Open Science Grid (OSG) which is an HTC resource funded by NSF and DOE. After discussing the project with Balamurugan Desinghu of the OSG User Support Team, Wim installed the CFM suite of programs and we were given user access to run our calculations. The table below shows the difference in computing times on CHPC vs OSG. Although CHPC was considerably faster on a per-job basis, the rate-limiting factor was the number of free nodes on Ember—this made OSG an ideal solution, where the entire batch was parallelized and ran to completion in ~12 hours.

	Job Size (230,737 moecules total)	***Run Rate***	***Free Nodes***	***Time to Completion***
Ember Cluster	462 files containing 500 molecules each	69 molecules / hour	Typically 10-20	334 hours (based on 10 free nodes)
OSG	4,614 files containing 50 molecules each	4 molecules / hour	Thousands	12 Hours

Workloads that work well in the OSG model are serial jobs that take less than 2GB memory and can complete in less than 12 hours. Additional details can be found at *http:// www.opensciencegrid.org/about/what-kind-of-computational-problems-fit-well-on-osg* . If you think you may have a workload that is a good fit and would like more information or assistance in getting started, please contact us at *issues@ chpc.utah.edu*. We look forward to working with you.

New at *www.chpc.utah.edu*

The CHPC website now has a FAQ section found at: *https://www.chpc.utah.edu/documentation/faq.php*

We will continue to add common questions and their responses in this location.

Centos Upgrade

The upgrade from CentOS6 to CentOS7 is nearly complete. All general clusters have been updated and we are now in the process of identifying remaining servers that need upgraded. If you have any issues with using CHPC resources since the OS upgrade, please report them to *issues@chpc.utah.edu*.

Center for
HIGH PERFORMANCE COMPUTING
THE UNIVERSITY OF UTAH

The University of Utah
University Information Technology
Center for High Performance Computing
155 South 1452 East, Room #405
SALT LAKE CITY, UT 84112-0190

Thank You for Using Our Systems!

Welcome to CHPC News!

If you would like to be added to our mailing list, please provide the following information and send via the CHPC contact methods listed below:

Name:
Phone:
Email:

Department or Affiliation:

Address:
(UofU campus or U.S. Mail)

Please help us continue to provide you with access to cutting edge equipment.

ACKNOWLEDGEMENTS
If you use CHPC computer time or staff resources, we request that you acknowledge this in technical reports, publications, and dissertations.

Here is an example of what we ask you to include in your acknowledgements:
"A grant of computer time from the Center for High Performance Computing is gratefully acknowledged."

Please submit copies or citations of dissertations, reports, pre-prints, and reprints in which the CHPC is acknowledged one of the following ways:

Electronic responses

Email: [redacted]

Fax: (801)-585-5366

Paper responses

U.S. Mail: 155 South 1452 East, Rm 405
Salt Lake City, UT 84112-0190

U Campus Mail: INSCC 405