EPJ Web of Conferences **245**, 03005 (2020) https://doi.org/10.1051/epjconf/202024503005
CHEP 2019

Distributed Computing Software and Data Access Patterns in OSG Midscale Collaborations

Pascal Paschos[1,*], *Benedikt* Riedel[2], *Mats* Rynge[3], *Lincoln* Bryant[1], *Judith* Stephen[1], *Robert* Gardner[1], *Edgar* Fajardo[4], *John* Hicks[5], *Frank* Wuerthwein[4], and *James* Clark[6]

[1]Enrico Fermi Institute, University of Chicago 955 E. 56th St. Chicago IL 60637
[2]University of Wisconsin-Madison 222 West Washington Ave, 500, Madison, WI, 53703
[3]Information Sciences Institute, University of Southern California, Los Angeles, CA 90007
[4]University of California San Diego 9500 Gilman Dr. La Jolla CA 92093
[5]Internet2 100 Phoenix Drive, Suite 111 Ann Arbor MI 48108
[6]Center for Relativistic Astrophysics and School of Physics, Georgia Institute of Technology, Atlanta, GA 30332

Abstract. In this paper we showcase the support in Open Science Grid (OSG) of Midscale collaborations, the region of computing and storage scale where multi-institutional researchers collaborate to execute their science workflows on the grid without having dedicated technical support teams of their own. Collaboration Services enables such collaborations to take advantage of the distributed resources of the Open Science Grid by facilitating access to submission hosts, the deployment of their applications and supporting their data management requirements. Distributed computing software adopted from large scale collaborations, such as CVMFS, Rucio, xCache lower the barrier of intermediate scale research to integrate with existing infrastructure.

1 Introduction

In Open Science Grid (OSG) [1][2] we support science at all scales. The experiment specific workload of large scale science projects requires significant software and computing technical teams. Such teams are responsible for the development, validation, and deployment of the computational workload and data management systems and coordinate closely with OSG technical teams in integration and operationalization across the grid sites. On the other end of the scale are the individual researchers across multiple fields, who access the grid opportunistically for their job submissions. Their workflows are often supported by OSG research facilitators with expertise in high-throughput computing. Some collaborations fall in between the large science projects and the individual researchers. We define that region of scale as "Midscale" and it typically involves dozens of researchers, spread across several institutions working on the same project. Their needs scale up to several petabytes per year with often episodic CPU utilization.

Medium-scale scientific organizations can either use opportunistic resources in the OSG or run on allocated distributed resources provided by participating institutions. Opportunistic access can be gained by leveraging the OSG Connect/CI Connect infrastructure at the University of Chicago (UChicago), a number of login nodes provisioned

*e-mail: paschos@uchicago.edu

as submission nodes for general purpose or collaborative projects. Allocated resources can be added to the OSG pool for the use of the collaboration via a Hosted CE (Compute Element) solution (https://opensciencegrid.org/docs/compute-element/hosted-ce/) or by a CE deployed on-campus infrastructure (https://opensciencegrid.org/docs/compute-element/htcondor-ce-overview/). Members of the collaboration can then authenticate to the allocated resources by generating short-lived proxy certificates which are included in the HTCondor submit scripts.

The limited availability of technical teams embedded within the Midscale collaborations presents challenges for support services. The latter will need to provision a distributed computing environment with easy access to software, support of containerized solutions, and enable reliable distribution of data and automation. In this proceedings paper, we list a few distributed cyberinfrastructure solutions that were adopted by OSG from large scale projects, for example, the LHC, and then outline their implementation in specific midscale collaborations.

2 Examples of Distributed Software

2.1 CVMFS

CERN Virtual Machine File System (CVMFS) [3] is a POSIX-like read-only file system in userspace (FUSE) that delivers a portable solution in making files available on-demand to jobs running on a distributed infrastructure. CVMFS has been commonly used by the Worldwide LHC Computing Grid (WLCG) [4] to efficiently distribute scientific software and small files over HTTP/HTTPS with several layers of caching. CVMFS has been adopted by the Open Science Grid as the main means to provide repositories of application software and compiled research code at job runtime. The OSG's central repository, OASIS, can host both OSG software and external user/collaborative repositories supplied by Virtual Organizations (VO). OASIS consists of a CVMFS server (origin or Stratum-0), a replica of stratum-0 (stratum-1) and a node that VOs can use to publish updates for their repository. Alternatively, collaborations can provision their own origin servers and publish updates of their repositories to CVMFS. The use of outbound HTTP connections is managed by an HTTP server, namely Frontier-Squid, which was re-purposed from another experiment namely the CDF experiment at Fermilab.

The initial scope of CVMFS was to distribute software across a federated computing ecosystem with heterogeneous resources, making applications available in a publicly accessible namespace. Data intensive science and the expectation to secure access to proprietary data has extended the scope of CVMFS to include the capacity to service and deliver to remote sites scientific datasets collected from experiments (files over 1 GB) [5]. This motivated the deployment of the StashCache infrastructure, discussed in 2.2, which caches data distributed from an origin server and negated the requirement for compute sites to maintain full copies.

2.2 StashCache

OSG's caching infrastructure is based on SLAC's XRootD server and XRootD protocol [6]. The service leverages file block caching technology that uses servers and redirectors. The Cache Server is placed at several strategic geographical locations across the OSG national network. Internet2 (I2) provides the physical hardware and the network layer for a part of the caching network while OSG manages the deployment of StashCache endpoints and origins. StashCache [7] was born out of the need to only read the data once from the origins (usually

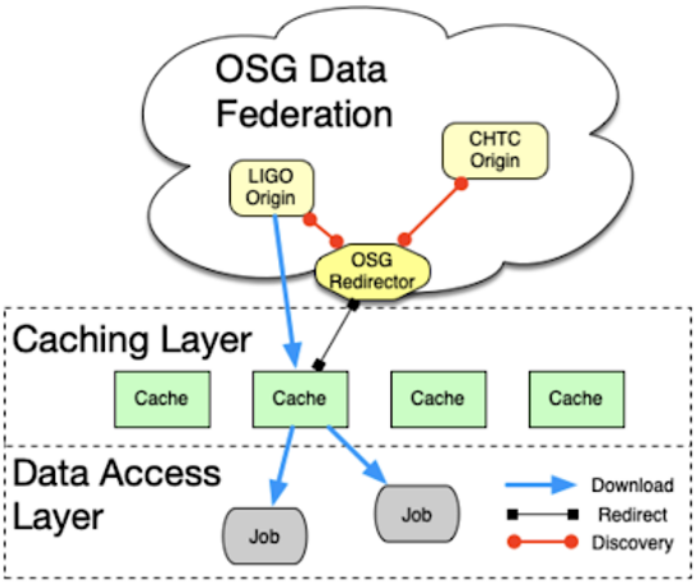

Figure 1. StashCache architecture for application performance in IGWN. Jobs running on the grid request data from the nearest Cache. Missing data on the caching later will be served directly by the Origin as the redirector - run by OSG Operations - discovers which Origin holds the data.

far away, mass storage services) and read it from a "nearby" cache. The implementation allowed for science applications, with needs to repeatedly access data, to not having to transfer the input files from the Origin server every time a job was submitted for execution.

Jobs utilize GeoIP in order to determine the nearest cache location. In the event a cache is unavailable to serve data to the job, the next available cache is queried. If the data is missing, then the Origin can serve the data. Once a client asks for a file, the redirector queries the servers bellow the tree if they have the file and, if they do, the client is redirected to start a connection with the correct server. If none of the servers have the file the redirector asks the redirector above it. This kind of tree structure is called a federation, like CMS Federation: Anydata, Anytime, Anywhere (AAA) [8].

Jobs communicate to the cache and transfer data using HTTP(s) via CVMFS. StashCache sits in front of a federation, therefore when a job at a site asks for a file, StashCache acts as a client to gather the file from the origin(s), serve it from memory to the worker node and then queue it to be saved on a local disk. The architecture can be seen in Figure 1. In order to provide the access to the data over CVMFS StashCache is configured to provide not only access using Xrootd protocol but also HTTP(s), since CVMFS can only retrieve files using HTTP.

2.3 Rucio

Rucio [9] was developed for the ATLAS experiment in order to manage the distribution of data across the collaboration's member sites. It is an open-source software framework that provides a scalable solution in organizing and managing data access. Since ATLAS, Rucio

has been adopted or evaluated for the use of other experiments such XENON, CMS, Icecube, LIGO, DUNE and others.

Rucio provides an interface to heterogeneous computing resources and an integration with the workflow management system chosen by the collaboration. Rucio will ensure that requested files are available at their destinations and new ingested files are registered and ready to be distributed. Access to the variety of supported storage systems, including tape, is authenticated via X.509 certificates or ACLs (access control lists). The File Transfer System (FTS) is used to provide third party copy over the network between storage sites. Rucio submits requests to FTS to transfer files and monitors progress into order to report on successful completion or errors. Rucio is not bound to a particular file transfer protocol. The protocol used can depend on the client accessing the data.

Data Identifiers (DIDs) are used to organize the ingested data with levels of granularity (file, dataset, container) and a standardized naming scheme, a scope and a name, unique for each data object. The physical location of the DIDs, otherwise known as replicas, is defined in Rucio and is associated with a Rucio Storage Element (RSE). The configuration of each RSE contains all attributes necessary to access the local storage which enhances the flexibility of the service delivery. The namespace of data uploaded into Rucio can either be managed automatically by Rucio (Deterministic RSE) or by explicitly providing the path to file on the storage endpoint (non-Determinisitic RSE).

UChicago maintains the Rucio server for the XENON experiment along with the deployment of the storage elements (RSE) and the File Transfer Service (FTS). We will discuss in more detail the deployment for the XENON experiment in Section 3.1 and report on the operational experience from production usage.

3 Examples of Midscale Collaboratios

3.1 XENON1T/NT

The new phase for the XENON experiment, XENONnT, prepares for the start of operations in 2020 to continue the search for Dark Matter at the Laboratori Nazionali del Gran Sasso (LNGS) facility in Italy. This phase of the experiment features higher detector sensitivity and two times the volume of expected experimental data when compared to the previous phase (1T). For 1T, the collaboration extensively leveraged OSG compute resources with the support of teams from University of Chicago to process the raw data from the detector into minitree ROOT files for analysis. Over 15 million compute hours were consumed on OSG for XENON1T in 24 million jobs. In Figure 2, we show a schematic of the infrastructure that supports grid job submissions to the OSG for the XENON collaboration. Besides being a gateway to the OSG compute sites, the infrastructure is also the central management hub for the Rucio data management services.

Figure 3 shows the layout of the data flow for XENONnT. Data from the detector (below ground) are processed by the streaming analysis framework STRAXEN [10] before ingested by Rucio and distributed to the various endpoints of the collaboration. STRAXEN was based on the generic STRAX analysis framework [11] and modified for XENON data. STRAXEN is modular; components of the pipeline can be used as stand alone tools at any stage in the processing chain which greatly accelerates the processing of the raw data from the detector when compared to the previously used workflow PAX. As a result, we expect that most of the data will be stream-processed - online - before they are ingested by Rucio and distributed to the various Rucio Storage Elements (RSEs). A major change for nT is that both processed and raw data will be managed by Rucio where 1T only stored the raw data. Further analysis

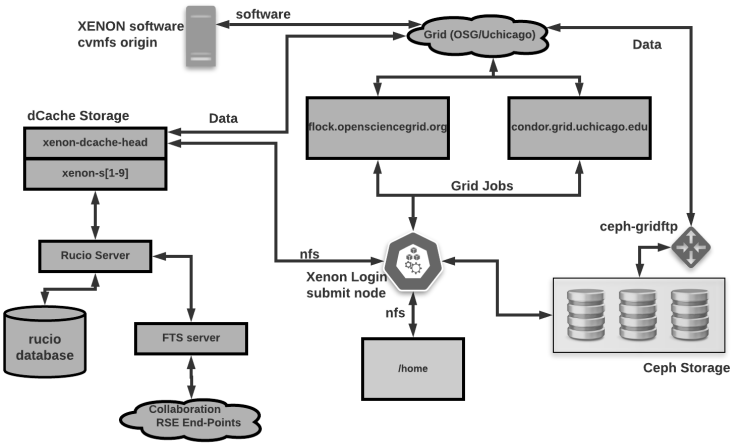

Figure 2. OSG infastructure at UChicago which supports the XENON experiment. The deployment has two components. The first component is the hosting of the Rucio/FTS server, Rucio DB and the XENON data for the UC_OSG_USERDISK Rucio Storage Element (RSE) on dCache storage. The second component is the submission host - login node - for job submissions to the OSG. In addition to this infrastructure, UChicago provides access to the HPC resource - Midway - for analysis and storage - DaLI.

of the data for science results will continue to rely on the Midway HPC resource at UChicago as it was for 1T.

The raw XENONnT data will be archived on tape, with the UChicago RSE (UC_OSG_USERDISK) keeping an almost complete copy on dCache. This pattern of storage is similar to the current one for 1T shown in Figure 4. Grid processing - submitting jobs to the OSG - will still be required for either re-processing campaigns or as an alternate method if there are unexpected failures or limitations in the online mode at the site of the experiment. Grid processing will be handled by *Outsource* [10], a complete computing environment setup that is integrated with the Pegasus Workflow Management System (WMS) [12]. The computing environment for XENONnT is containerized and deployed to the remote compute sites as a Singularity image [13]. UChicago hosts the software CVMFS origin for the collaboration, used to publish updates of the repository. The Singularity container for XENONnT includes the analysis software chain in addition to tools that upload the results back into Rucio. A new tool, called aDMIX (advance Data Managment in XENON) [10] replaces Ruciax used during the 1T phase. aDMIX is built on top of the Rucio Command Line Interface (CLI) client and provides an API to manage updates of the Rucio catalogue with the results from the processing jobs.

The computing and storage model for XENON is mostly federated. Computation for the purpose of processing low-level data, in the form of High Throughput Computing (HTC) workflows, is distributed across the grid and facilitated by the availability of software over CVMFS. Analysis computations, to produce high-level science grade results, is carried out on a single HPC resource (Midway at UChicago) mainly due to the availability of high memory nodes. Data are distributed and managed across multiple sites by the Rucio data management solution adopted from the ATLAS experiment. Additional HPC resources can be leveraged for analysis, however there are still challenges in deploying CVMFS on HPC clusters; critical

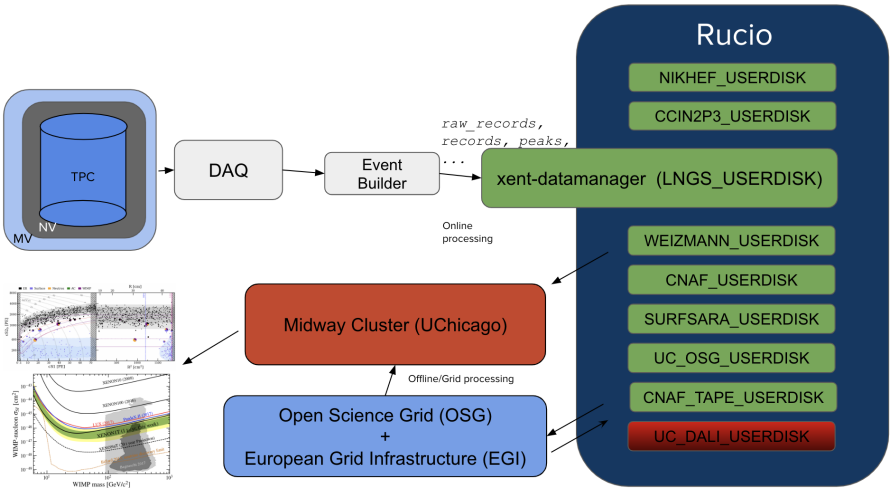

Figure 3. Data management workflow for XENONnT. The introduction of STRAX/STRAXEN for the on-line processing chain of the raw data accelerates the path to the Midway HPC resource and minimizes the need for Grid processing on OSG and EGI

Figure 4. Rucio RSE utilization for the XENON1T data. During the 2 years of operations, the experiment has produced over 800 TB of data which are managed by Rucio. Top: Number of files in each RSE endpoint. Bottom: Storage utilization in TB for each RSE endpoint

in making the analysis tools available there. Data centers are reluctant to support the deployment of CVMFS due to restrictive policies on worker nodes, lack of outbound connectivity or the absence of local disks which provide the local caching layer used by CVMFS. The latter can be deployed on a HPC cluster if Singularity is made available there. This allows for the use of the CVMFS FUSE client from inside the container via the pre-mount feature of the libfuse3 library [14], a strategy employed on the Midway HPC cluster.

3.2 VERITAS

VERITAS (Very Energetic Radiation Imaging Telescope Array System) is a ground-based instrument studying gamma-rays in the GeV to TeV range. The array of four 12 m optical reflectors at the Fred Lawrence Whipple Observatory (FLWO) in southern Arizona images Cherenkov radiation produced when the gamma rays interact with the Earth's upper atmosphere.

The VERITAS collaboration includes some 100 members in 20 institutions around the world and has been extensively leveraging OSG infrastructure for processing the signal chain from the instrument along with a large volume of Monte Carlo simulations. The collaboration consumed over 2 million CPU hours (over 5 million jobs) on OSG during the past year as shown by the GRACC accounting system in Figure 5.

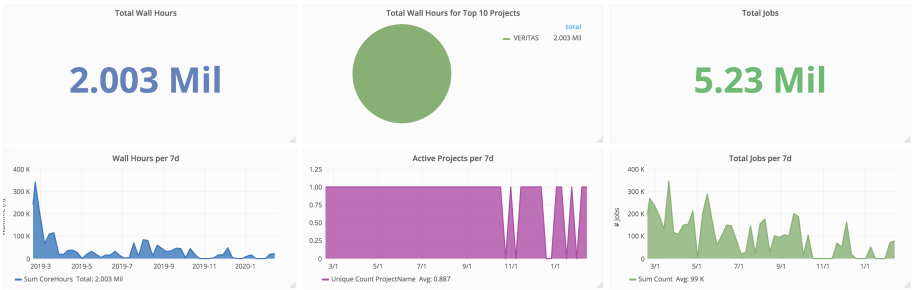

Figure 5. The VERITAS project consumed over 2 million compute hours on OSG over the past year and submitted over 5 million condor jobs to the grid.

VERITAS is primarily supported by OSG midscale collaboration services via a dedicated login node on the CI-Connect infrastructure as the HTCondor job submission point to the OSG (Figure 6). In addition, the collaboration leverages available disk on OSGconnect stash storage service, a mounting point for UChicago's CephFS storage solution. To date, VERITAS has collected over 400 TB of processed data on stash. OSG midscale collaboration services also supports the software repository for VERITAS. A CVMFS origin server is provisioned and used to publish and distribute updates to all grid compute sites which have CVMFS mounted.

The Pegasus Workflow Management System (WMS) is integrated in the VERITAS pipeline for job submissions to the OSG and other UChicago resources by provisioning two separate flocking services connected the submission node, thus allowing for flexibility in targeting available slots. However, the VERITAS pipeline also produces a large number of temporary files at runtime which exceeds the local storage capacity of typical OSG grid compute sites. In order to address such a requirement, the UChicago team provisioned a dedicated RADOS gateway [15] that allows grid jobs to access CephFS for ephemeral storage. A RADOS gateway is built on top of librados to provide an AWS S3 compatible block storage functionality via a RESTful API interface.

VERITAS does not actually analyse telescope data on the OSG. It uses OSG to run simulations of particle showers for gamma-ray reconstruction. Data from the instrument are hosted in an archive at UCLA and raw data analysis takes place at Georgia Tech's PACE cluster. The simulations on the OSG use the CORSIKA package [16] to simulate the air shower of electron/positron pairs induced by the interaction of a very high energy gamma-ray with the Earth's atmosphere. A package called GrOptics then reconstructs the paths of the gamma-ray photons. This process is repeated for hundreds of millions of showers.

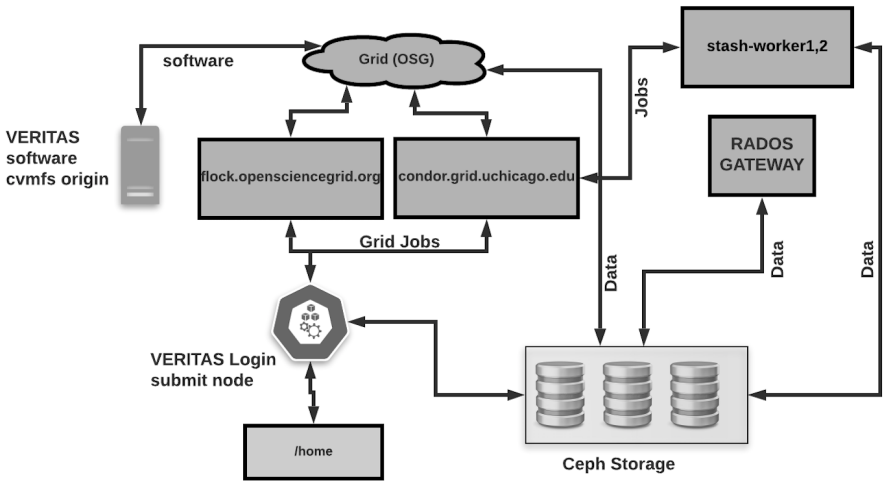

Figure 6. Connect infrastructure deployed at UChicago for the VERITAS collaboration.

Tree merging of the individual shower events is the last stage in the pipeline and it is memory intensive enough (> 6 GB per job) to exceed the typical available memory per core on OSG compute sites. Therefore, two dedicated high memory nodes (stashworkers) have been provisioned for the collaboration that are treated like another OSG endpoint. The stashworker nodes can be targeted by the HTCondor submission script and have OSG stash NFS-mounted which allows for access to the project's data on Ceph. A graphical schematic of the VERITAS infrastructure deployed at UChicago is shown in Figure 6.

The computing and storage model for VERITAS is mostly federated in compute but centralized in storage. Data reduction takes place on an institutional cluster while simulation jobs, ideally suited for OSG, are distributed to the grid along with software from a CVMFS origin. Storage of the simulated data are housed on the OSGConnect stash storage with instrument data on a tape at UCLA. VERITAS is an example of a midscale collaboration that has adequate local resources for their core pipeline but requires additional capacity and ephemeral storage to accommodate workflows on the grid in order to accelerate the path to discovery.

3.3 South Pole Telescope - SPT3g

SPT3g is an enhanced capability camera installed on the 10 m South Pole Telescope for Cosmic Microwave Background (CMB) observations by conducting wide-field millimeter and sub-millimeter surveys. The camera, installed in 2017, provides researchers with ten times more sensors compared to the previous phase SPTpol. The project searches for galaxy clusters using the Sunyaev–Zel'dovich effect [17], which describes the scattering of CMB photons by the electrons in the hot gas found in the intergalactic medium.

In Figure 7 we depict the infrastructure setup that supports the data flow and storage for the experiment. Raw data from the instrument at the South Pole are uploaded via a satellite link to facility at the United States Antarctic Program portal (USAP). A buffer machine at UChicago receives the data from USAP via an SFTP connection. Storage allocations at the HPSS archive at NERSC and on DaLI at the Research Computing Center (RCC) at Uchicago

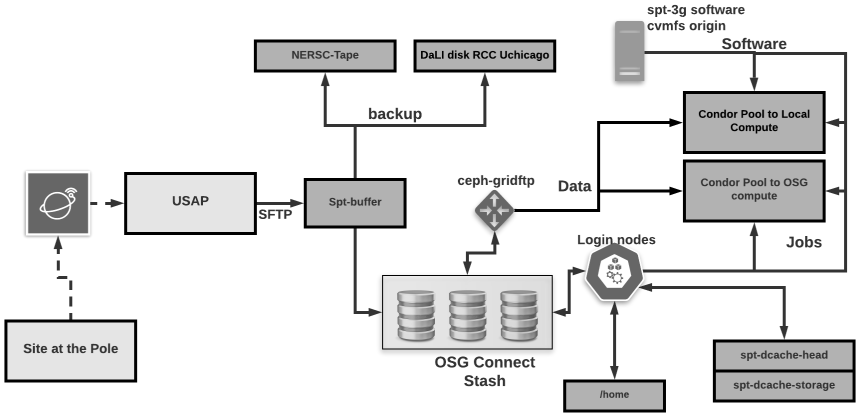

Figure 7. Infrastructure supporting research for the South Pole Telescope collaboration. Data from the instrument at the South Pole are uplinked/downlinked via a satellite to the USAP facility and then transferred to UChicago for processing, backup and distribution.

are used for backups of the raw data via Globus-GridFTP transfers from cron jobs on buffer. The raw data on the buffer machine can then be pre-processed via jobs on the OSG and the production quality results stored on the Connect Stash storage for further analysis by the SPT researchers. Researchers can use two login nodes provisioned as submit hosts to the Open Science Grid and a GridFTP door to move data out of Ceph storage to the grid compute sites. The software stack for SPT is maintained and updated at the same stratum-0 machine that serves other collaborations and distributed over CVMFS.

SPT has adopted a distributed computing model as pre-production and production jobs are submitted to the OSG. Storage of active data is centralized on stash, with NERSC and RCC keeping backup copies. Although similar to VERITAS, in the SPT case there is a continuous inflow of data from the experiment and that creates challenges in a centralized storage model. In Figure 8 we show the growth of the storage utilization by SPT data on CephFS. Due to limited capacity, there is a need to routinely purge older data to make room for fresh datasets. The most consequential issue is that the growth in the volume of accumulated data from the telescope also requires growth in the volume of grid jobs submitted to process them. This places strains in the available bandwidth as access to data on Ceph needs to be proxied through the GridFTP door which is limited to 20 Gbps (2×10 Gbps). This, in turn, limits the number of concurrent jobs that can be submitted to the OSG, when users transfer the data out of Ceph using the GridFTP protocol. To mitigate the bandwidth issue, SPT data are at present being moved to dCache storage. This promises increased bandwidth, as each dCache server daemon implements their own GridFTP protocol and does not require a door. Even though the expected bandwidth boost to 80Gbs should help increase performance, the issue of limited storage capacity will remain for SPT unless a federated storage model is adopted.

3.4 IGWN

The International Gravitational-Wave Observatory Network (IGWN) is the US-based LIGO, European based VIRGO and KARGA based in Japan. IGWN jobs used to read data remotely from the origin using CVMFS, however the collaboration's computational workflows mainly

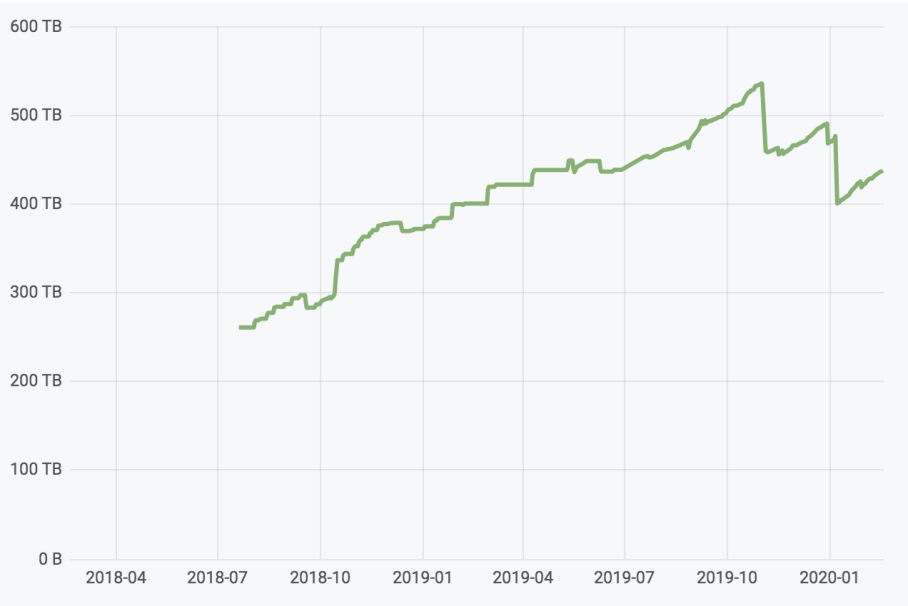

Figure 8. Growth of central storage utilization for the South Pole Telescope collaboration over the past 2 years. Due to limits on the capacity of the connect stash storage (Ceph), growth is caped and data purges are required to allow for processing of more recent data from the Pole.

focused on parameter estimation workflows which require reading the same set of data repeatedly. This inspired the idea to use a caching layer in order to provide proximity of frequently used datasets to compute sites.

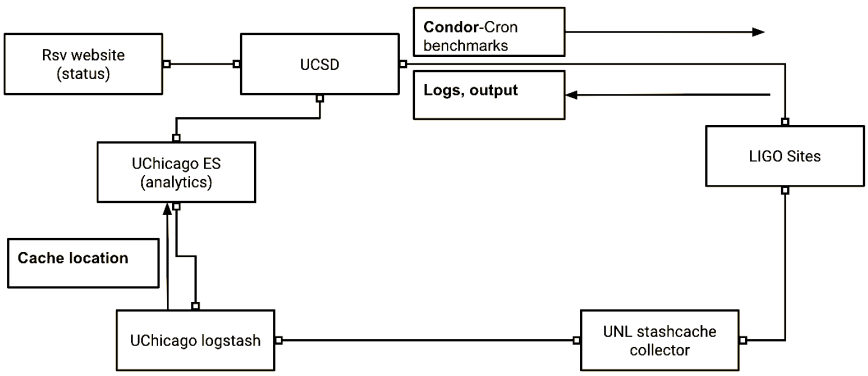

Figure 9. Infrastructure to measure dependency of IGWN job efficiencies to StashCache node availability and proximity to the compute site. Data are indexed in an Elastic Search instance at UChicago and displayed over a Kibana front-end.

The IGWN collaboration, uses dedicated login nodes distributed across the US, Europe and Japan to authenticate users and submit jobs to the grid. It leverages the StashCache data federation described in 2.2, which caches data near compute sites for faster delivery

to jobs requesting them from CVMFS. At present, the StashCache federation consists of 7 sites, 6 in the US and 1 in Europe (Amsterdam). Five of the sites have been deployed in collaboration with Internet2 and managed by OSG operations. We refer to Fadarjo et al on the *Worldwide deployment via Kubernetes* of the StashCache infrastructure also published in these proceedings.

The IGWN collaboration distributes both software and data (Frames) over CVMFS. While the software repository (/cvmfs/oasis.opensciencegrid.org/ligo/sw) and past frame files are publicly available, frame files from current observing runs (/cvmfs/ligo.osgstorage.org/frames) require user authentication for access. The collaboration uses Pegasus WMS to submit jobs and mainly run the following applications on the OSG: Bayeswave [18], used for distinguishing gravitational wave signals from noise irrespective of the waveform morphology. PyCBC [19] is a parameter estimation code used to analyze gravitational-wave data in order to find signals. LALsuite [20] is a collection of gravitational wave data analysis codes. These three software packages are well suited for HTC jobs on OSG and typically ramp up usage of grid resources towards the end of an observing run period. In addition, a new algorithm for parameter inference of gravitational wave sources called RIFT [21] [22] has been implemented to use CUDA kernels for accelerated performance. IGWN users can access GPUs and run RIFT on the Pacific Research Platform (PRP) which is a Kubernets managed resource and provides access to about 330 GPUs.

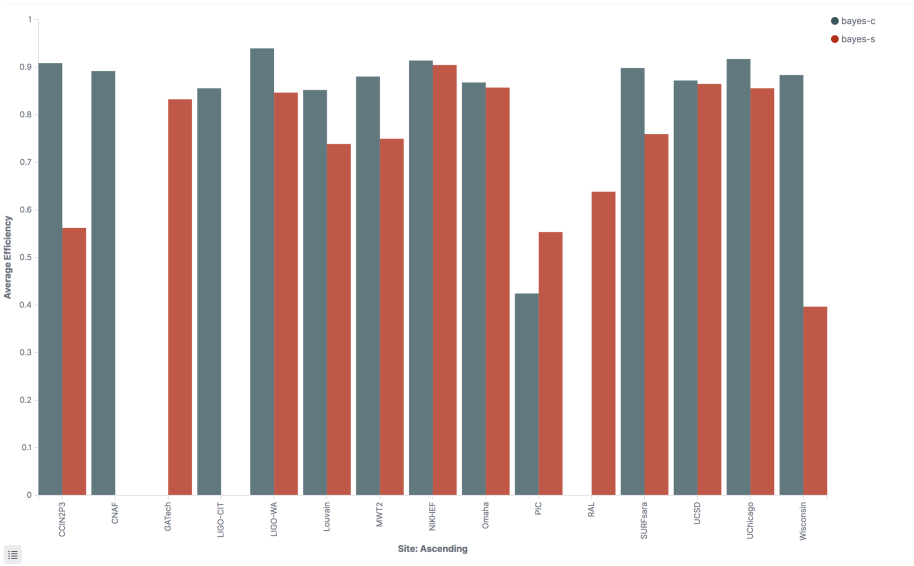

Figure 10. Benchmark Monitoring of job efficiencies across IGWN compute sites. Shown here is the average job efficiency across sites for two Bayeswave benchmarks: Code deployed through a conda-python virtual environment (bayes-c) and code deployed through a Singularity container (bayes-s).

In order to evaluate the impact of the caching infrastructure for IGWN, shown in Figure 1, we are setting up an infrastructure monitoring service that ingests performance data from benchmarks that run LIGO type workflows. A total of 8 benchmarks at present are submitted to OSG from a host at UC San Diego (UCSD) using OSG's Resource and Service Validation (RSV). Condor_cron is used to submit every 4 hours a sequence of benchmark jobs to the grid and collect results from the HTCondor history ClassAd files, such as job CPUtime and Walltime, shown in Figure 9. The measured efficiency data are then period-

ically ingested, indexed into an Elastic Search analytics platform and displayed through a Kibana interface [23]. In Figure 10, we show data from two benchmarks collected through this process. Both benchmarks are a test implementation of the Bayeswave application, read from O3-cycle data in /cvmfs/ligo.osgstorage.org/frames and are continuously submitted to read different frames of the same size. In doing so, we force reading frames from StashCache instead a locally cached copy that was served on the site during the previous job. At present, we are working towards correlating the ClassAd data from the HTCondor history files with information aggregated by the StashCache collector at the University of Nebraska-Lincoln (UNL) which keeps track of the read access of the caching infrastructture by individual grid jobs.

3.5 Other Midscale collaborations

The KOTO experiment at the Hadron Experimental Facility of Japan Proton Accelerator Research Complex (J-PARC) site, searches for new physics that breaks the CP (charge/parity) symmetry by studying a rare neutral-kaon decay channel. The next Enriched Xenon Observatory (nEXO) is a Lawrence Livermore National Laboratory (LLNL) particle physics experiment to be carried out at SNOLAB Canada. It searches for neutrinoless double beta deday of Xe-136.

Both experiments have a set of common challenges in their data processing campaigns; the primary computing center (LLNL for nEXO, KEK-CC for KOTO) does not have the computational capacity to accommodate the large volume of simulated event calculations. The latter are Monte-Carlo simulations computed with the widely used GEANT4 toolkit - used in High Energy Physics experiments. HTC on the OSG is perfectly suited as a computing cyberinfrastructure ecosystem for these types of calculation. Both nEXO and KOTO use OSG for their HTC calculations, with their software stack distributed over CVMFS from a Stratum-0 server at UChicago or OSG's OASIS.

However, it would be advantageous to those collaborations if their OSG and their facility allocations are joined as an integrated computing target to facilitate job submissions to both from a single submission point. Such an approach allows for researchers to be unconcerned with individual resource availability, heterogeneity, different scheduling environments, and priorities. One solution is to use Virtual Clusters for community computation (VC3) [24] which allows binding allocations from individual investigators across institutions. VC3 sets up a single submit host and provisions a dynamically sized number of middleware worker nodes by pooling together resources available at the individual institutions. Users schedule jobs to the virtual worker nodes and are provided with a uniform application software stack to run on the remote nodes.

VC3 has been evaluated for other collaborative projects such as CMS and Icecube and it's not without implementation challenges. For example, connecting resources secured with multi-factor authentication requires a special arrangement at the institutional endpoint (whitelisting the VC3 submission host). The solution is mentioned here because of two reasons. First, results so far for the KOTO collaboration have been encouraging enough to expand this to nEXO. The computing schema that VC3 supports bonds well with the needs of these collaborations and further work is planned to set up to production pipelines. Second, VC3 itself has re-purposed other distributed software, such as AytoPyFactory (APF), to deploy middleware workers to remote clusters that HTCondor supports. APF was developed at Brookhaven National Laboratory (BNL) to manage job submissions for the ATLAS LHC experiment.

4 Conclusions

Distributed software and tools built for LHC experiments have been re-purposed for other collaborations that use OSG for a significant portion of their computational workflow. Deployment and site availability of tools such as CVMFS, Rucio, FTS, and XRooD facilitate computational and data management at scale for multi-institutional science collaborations and provide a path forward for researchers to leverage the OSG more efficiently and thus be more productive. Midscale collaborations rely on OSG to provide an environment that minimizes implementation and adoption overhead and lowers the bar for research to integrate their domain-specific workflow modalities to the grid. There are operational challenges, establishing and documenting best practices in coordinating and meeting expectations when interfacing with heterogeneous computing, storage resources and policy guidelines such as HPC institutional data-centers. The biggest two challenges is the absence of dedicated staff with expertise in grid computing on the collaboration side and coordinating with institutional resource administrations. Both of these can limit support due to human bandwidth and contending priorities in the delivery of service.

The first challenge can be met with training researchers in best practices. In doing so, they can be empowered to provide feedback to service developers and operators in OSG on how to best support multi-institutional research. Also, collaborations can benefit if OSG collaboration support is engaged as early as the proposal stage to help guide integration to grid resources and advise on feasibility. Even though there is a common core of software and infrastructure solutions that can be deployed for their use, each case has unique challenges and it helps to identify them early on.

The second challenge can be met as part of an on-going conversation on the future of a federated national cyberinfrastructure. Federal funding agencies strongly encourage multi-institutional collaboration and coordination in leveraging national wide resources such as the OSG. An alignment between the priorities of local data centers and grid infrastructure can be achieved as part of the common goal to facilitate computational research and the path to discovery and innovation.

Acknowledgements

This research was performed using resources provided by the Open Science Grid [1][2], which is supported by the National Science Foundation award 1148698, and the U.S. Department of Energy's Office of Science.

This research used the Pegasus Workflow Management System funded by the National Science Foundation under grant #1664162.

This research was also supported by NSF's Cyberinfrastructure for Emerging Science and Engineering Research (CESER) grant in Multi-Messenger Astrophysics #1841479

References

[1] R. Pordes, D. Petravick, B. Kramer, D. Olson, M. Livny, A. Roy, P. Avery, K. Blackburn, T. Wenaus, F. Würthwein et al., *The open science grid*, in *J. Phys. Conf. Ser.* (2007), Vol. 78 of *78*, p. 012057

[2] I. Sfiligoi, D.C. Bradley, B. Holzman, P. Mhashilkar, S. Padhi, F. Wurthwein, *The pilot way to grid resources using glideinWMS*, in *2009 WRI World Congress on Computer Science and Information Engineering* (2009), Vol. 2 of *2*, pp. 428–432

[3] P. Buncic, C.A. Sanchez, J. Blomer, L. Franco, A. Harutyunian, P. Mato, Y. Yao, *CernVM a virtual software appliance for LHC applications, in J. Phys. Conf. Ser.* (2010), Vol. 219, p. 042003

[4] I. Bird, *Computing for the Large Hadron Collider, in Annual Review of Nuclear and Particle Science* (2011), Vol. 61, pp. 99–118

[5] D. Weitzel, B. Bockelman, D.A. Brown, P. Couvares, F. Würthwein, E.F. Hernandez, *Data Access for LIGO on the OSG*, in *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact* (2017), PEARC 17, pp. 1–6

[6] L. Bauerdick, K. Bloom, B. Bockelman, D. Bradley, S. Dasu, J. Dost, I. Sfiligoi, A. Tadel, M. Tadel, F. Wuerthwein et al., *XRootd, disk-based, caching proxy for optimization of data access, data placement and data replication, in J. Phys. Conf. Ser.* (2014), Vol. 513

[7] D. Weitzel, M. Zvada, I. Vukotic, R. Gardner, B. Bockelman, M. Rynge, E. Hernandez, B. Lin, M. Selmeci, *StashCache: A Distributed Caching Federation for the Open Science Grid, in Proceedings of the Practice and Experience in Advanced Research Computing* (2019), PEARC 19, pp. 1–7

[8] K. Bloom, the CMS Collaboration, *CMS Use of a Data Federation, in J. Phys. Conf. Ser.* (2014), Vol. 513, p. 042005

[9] M. Barisits, T. Beermann, F. Berghaus, B. Bockelman, J. Bogado, D. Cameron, D. Christidis, D. Ciangottini, G. Dimitrov, M. Elsing et al., *Rucio: Scientific Data Management, in Computing and Software for Big Science* (2019), Vol. 3, p. 11

[10] XENON Collaboration, *Software for the XENONnT experiment* (2019), `https://github.com/XENONnT`

[11] XENON Collaboration, *Streaming analysis for xenon experiments* (2019), `https://github.com/AxFoundation/strax`

[12] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P.J. Maechling, R. Mayani, W. Chen, R. Ferreira da Silva, M. Livny et al., *Pegasus: a Workflow Management System for Science Automation, in Future Generation Computer Systems* (2015), Vol. 46, pp. 17–35

[13] G.M. Kurtzer, V. Sochat, M.W. Bauer, *Singularity: Scientific containers for mobility of compute, in PLOS ONE* (Public Library of Science, 2017), Vol. 12, pp. 1–20

[14] *CernVM-FS Documentation* (2019), `https://cvmfs.readthedocs.io`

[15] *RADOS GATEWAY*, `https://docs.ceph.com/docs/bobtail/radosgw`

[16] D. Heck, J. Knapp, J. Capdevielle, G. Schatz, T. Thouw, *CORSIKA: A Monte Carlo code to simulate extensive air showers, in Forschungszentrum Karlsruhe Report FZKA 6019* (1998)

[17] R. Sunyaev, Y. Zel'dovich, *The Interaction of Matter and Radiation in a Hot-Model Universe, Astrophysics and Space Science* (1969), Vol. 4, pp. 301–316

[18] T.B. Littenberg, N.J. Cornish, *Bayesian inference for spectral estimation of gravitational wave detector noise, in Phys. Rev. D* (2015), Vol. 91, p. 084034

[19] C.M. Biwer, C.D. Capano, S. De, M. Cabero, D.A. Brown, A.H. Nitz, V. Raymond, *PyCBC Inference: A Python-based Parameter Estimation Toolkit for Compact Binary Coalescence Signals, in Publications of the Astronomical Society of the Pacific* (IOP Publishing, 2019), Vol. 131, p. 024503

[20] LIGO Scientific Collaboration, *LIGO Algorithm Library - LALSuite*, free software (GPL) (2018), `https://git.ligo.org/lscsoft/lalsuite`

[21] J. Lange, R. O'Shaughnessy, M. Rizzo, *Rapid and accurate parameter inference for coalescing, precessing compact binaries, in arXiv:1805.10457 [gr-qc]* (2018)

[22] D. Wysocki, R. O'Shaughnessy, J. Lange, Y.L.L. Fang, *Accelerating parameter inference with graphics processing units, in Phys. Rev. D* (American Physical Society, 2019), Vol. 99, p. 084026

[23] *Elastic Search*, `https://www.elastic.co`

[24] L. Bryant, J. Van, B. Riedel, R. Gardner, J.C. Bejar, J. Hover, B. Tovar, K. Hurtado, D. Thain, *VC3: A Virtual Cluster Service for Community Computation, in Proceedings of Practice and Experience in Advanced Research Computing* (2018), PEARC 18, pp. 1–8