Petrel: A Programmatically Accessible Research Data Service

William E. Allcock, Benjamin S. Allen, Rachana Ananthakrishnan, Ben Blaiszik,
Kyle Chard, Ryan Chard, Ian Foster, Lukasz Lacinski, Michael E. Papka, and Rick Wagner
Argonne National Laboratory

ABSTRACT

We report on our experiences deploying and operating Petrel, a data service designed to support science projects that must organize and distribute large quantities of data. Building on a high-performance 1.7 PB parallel file system and embedded in Argonne National Laboratory's 100+ Gbps network fabric, Petrel leverages Science DMZ concepts and Globus APIs to provide application scientists with a high-speed, highly connected, and programmatically controllable data store. We describe Petrel's design, implementation, and usage and give representative examples to illustrate the many different ways in which scientists have employed the system.

ACM Reference Format:
William E. Allcock, Benjamin S. Allen, Rachana Ananthakrishnan, Ben Blaiszik,, Kyle Chard, Ryan Chard, Ian Foster, Lukasz Lacinski, Michael E. Papka, and Rick Wagner. 2019. Petrel: A Programmatically Accessible Research Data Service. In *Practice and Experience in Advanced Research Computing (PEARC '19), July 28-August 1, 2019, Chicago, IL, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3332186.3332241

1 INTRODUCTION

Data-intensive science increasingly requires discipline-based data management tools, for example to distribute content from a curated repository, share files from a computational simulation or scientific instrument with collaborators, enable analyses of hosted data by collaborators, or accept uploads for analysis or publication. But would-be developers of such tools need a foundation on which to build: a foundation that is not necessarily provided by conventional research computing facilities, which are typically designed to support individual research projects rather than long-lived and community services. They need, in particular, storage systems that provide substantial capacity and high-speed storage and network access, and that can be managed and accessed entirely programmatically for easy integration into domain-specific workflows.

These considerations led the Argonne Leadership Computing Facility (ALCF) to establish the Petrel data service in 2015, initially as an experimental service to see whether and how people might use a programmatically-accessible storage service, and then—as success stories emerged—as a production service for the scientific community. The current Petrel system provides Globus access to 1.7 PB high-speed storage, connected to local and wide area networks at 100+ Gbps, and co-located with a 14-node Kubernetes cluster. Users can request allocations of 100 TB or more. If approved, they can then use Globus APIs (and associated Web interfaces) to move files to and from this allocated storage, organize files within this storage, and to authorize others to do the same. Petrel thus provides a configurable solution to the sharing of data with colleagues and the community, while at the same time keeping large datasets close to large computational resources.

We have found the results of this experiment to be highly encouraging. Dozens of groups have made use of Petrel in one way or another to manage petabytes of data and to transfer data between Petrel and more than 600 remote locations. Many groups have integrated its use into workflows, for example to automate distribution of data from experimental facilities and to organize and share data from simulation campaigns. Petrel thus represents a way in which a high-performance computing facility such as ALCF can usefully evolve beyond its traditional role of providing access to supercomputers for simulations [28] to become a highly usable service facility. Because Petrel is configured to integrate easily with user environments and workflows, it can serve as a hub for science communities and improve the usability and utilization of the core HPC center.

We report on the design and application of the Petrel system and reflect on lessons learned from its development and use.

2 THE PETREL SYSTEM

Historically, data at high-performance computing (HPC) centers have been located either on parallel file systems designed for rapid *internal* access or on data portal servers that support slower *external* data distribution; only the latter traffic was allowed to traverse the firewall. Thus high-speed data movement in and out of HPC centers was often difficult. Furthermore, collaboration in such environments has in the past been equally difficult, as users had to request or create accounts for each of their collaborators—a process that was often cumbersome and not flexible enough to support dynamic collaborations.

A recent reinvention of academic network architectures introduces the concept of a Science DMZ [13], in which specialized data servers are connected directly to high-speed networks and storage systems. This development has allowed for the emergence of the modern research data portal (MRDP) design pattern [8], in which control channel communications and data channel communications are separated, with the former handled by a web server computer deployed (most often) in the institution's enterprise network and the latter by specialized data servers connected directly to high-speed networks and storage systems. The MRDP design pattern also simplifies implementation of data portal functionality by outsourcing responsibility for managing data transfers, data access, and sometimes also authentication to external, often cloud-hosted, services, such as Globus. Thus data portal implementations can use simple REST API calls to manage data access.

PEARC '19, July 28-August 1, 2019, Chicago, IL, USA

ACM ISBN 978-1-4503-7227-5/19/07…$15.00
https://doi.org/10.1145/3332186.3332241

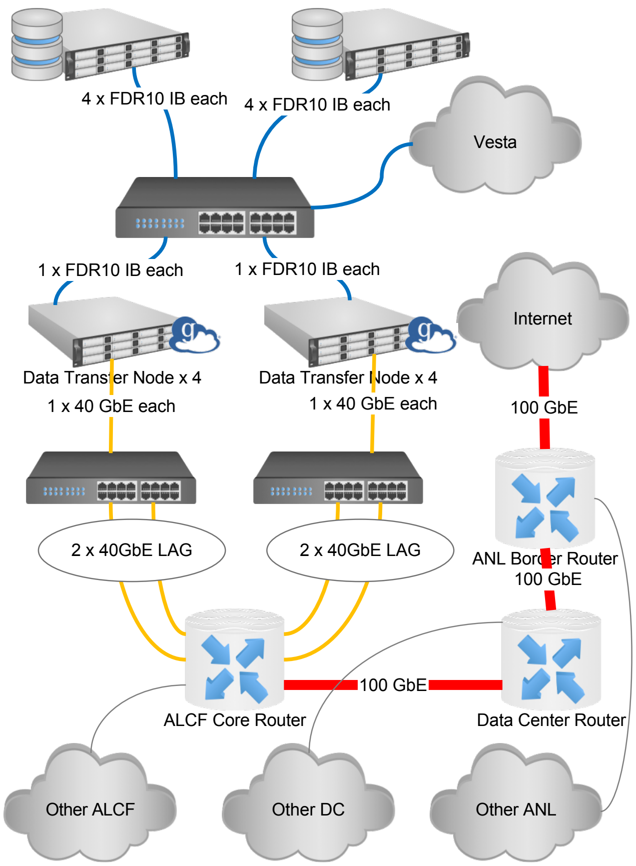

Figure 1: Petrel architecture, as described in the text. The NSD servers are at the top, with the DTNs are below them. The remainder of the figure shows how Petrel is connected to ALCF, other data center, and remote networks.

Petrel is designed to support the deployment and operation of applications that need to manage data movement through a highly connected, high-speed data store. It provides a user-oriented, collaborative storage model in which users manage their own isolated storage allocations including not only data organization but also dynamic sharing of files and directories within the allocation without requiring local user accounts. A central characteristic is its support for API access, which allows sophisticated behaviors to be implemented with modest amounts of programming. A second characteristic is its reliance on a Science DMZ [13] which provides specialized data servers that are connected directly to high-speed networks. In the following, we first describe the Petrel hardware and then the use of Globus services for remote data management and PetrelKube, a connected Kubernetes cluster for data analytics.

2.1 The Petrel Data Store and DTNs

The Petrel system comprises a parallel file system running the IBM Spectrum Scale file system (formerly known as the General Parallel File System, GPFS [23]), plus eight associated data transfer nodes (DTNs) [13] for remote access. This hardware is configured to operate as a Science DMZ, meaning that it is accessible from external networks without passing through the usual firewalls. Line-rate firewalls are in place, configured with network access control lists to limit access to Globus/GridFTP ports and, in particular, to limit control channel access to Globus service IP addresses.

The Petrel data store currently comprises a single IBM Elastic Storage Server (ESS) GL6. This device is configured with two IBM S822L Power8 based servers as IBM Spectrum Scale Network Shared Disk (NSD) servers, and one IBM S812L as a management and provisioning server. Each NSD server is connected with 4×FDR10 connections to the same fabric as the DTNs. Petrel includes six disk trays with a total of 348 6TB SAS HDDs. Configured with 8+2 parity and 3 drives worth of hot spare space, Petrel provides a usable capacity of 1.7P, and has been benchmarked via IOR at a maximum write rate of 16,563 MiB/sec (17,368 MB/sec) and maximum read rate of 23,111 MiB/sec (24,233 MB/sec).

Each of the eight Petrel DTNs has a single Mellanox ConnectX-4 40GbE NIC, a single Mellanox Connect-IB HBA (running at QDR), 64GB RAM (~42GB dedicated to Spectrum Scale), and a single socket Intel E5-1660 v3 8c @ 3.00GHz CPU. Both Mellanox cards sit on PCIe 3.0 x16 buses.

The eight Petrel DTNs are connected to two core Mellanox SX1710 36-port 40GbE switches maintained within Argonne's Joint Laboratory for Systems Evaluation (JLSE). The Petrel DTNs are split across the two switches, each connected with 1×40GbE. Each of the two 40GbE core switches has a 2×40GbE link aggregation group to the ALCF core router, which in turn has a 100GbE connection to the core router for the data center within which ALCF is located, which in turn connects at 100GbE to one of the ANL border routers. Thus, as Petrel traffic reaches beyond JSLE to ALCF, the data center, and the border, it shares bandwidth with an increasing range of other activities.

2.2 Globus Remote Access Services

Petrel builds upon Globus services [11] for identity and access management, data access and transfer, and data management. Specifically, it provides APIs for accessing, managing, and transferring data; as well as services for identity and access management. Petrel implements the MRDP design pattern [8], in which control channel communications and data channel communications are separated, with the former handled by a web server computer deployed (most often) in the institution's enterprise network and the latter by specialized data servers connected directly to high-speed networks and storage systems.

Petrel relies on Globus identity and access management services to support authentication, allocation management, and programmatic access to Petrel. By leveraging Globus Auth [27] users can login using one of many supported identity providers (e.g., institution, ORCID, Google). Each allocation is given a unique Globus Group [9] via which allocation administrators may manage the users who can access the allocation. Petrel's data access and management is provided via Globus APIs which are secured using Globus Auth. Thus, users may obtain secure OAuth 2.0 access tokens (via common OAuth authentication flows) and then present these tokens to access the APIs using user identities.

Access to Petrel is provided via Globus. Each of Petrel's eight DTNs is configured with Globus Connect Server software to enable high performance, third-party data transfer and secure HTTP data access. Each Petrel allocation is assigned a unique "shared endpoint"—a virtual endpoint on the Petrel host endpoint. The Globus Group is associated with the shared endpoint, enabling the owner and members of the allocation to access and manage data on the shared endpoint. Globus Sharing operates entirely with Globus Auth identities, that is it does not require that administrators or users of a shared endpoint have local accounts.

2.3 PetrelKube

A common request throughout Petrel's lifetime has been for co-located compute capacity "near" to Petrel's data storage. While Petrel already is tightly connected to ALCF supercomputing resources, the aim here is to provide support for mixed workloads and specifically coscheduled services not easily supported in HPC centers. Examples of such workloads include, applying analytics capabilities, operating NoSQL databases (MongoDB, Elasticsearch, etc.), running machine learning inference services, and batch processing of data after collection by instruments, and long running, persistent services.

In 2018, to explore these ideas, we created PetrelKube—a testbed Kubernetes cluster that provides connected analytics capabilities to Petrel. PetrelKube comprises 14 nodes, each with two E5-2670 CPUs, 128GB RAM, two 300GB hard drives in RAID 1, two 800GB Intel P3700 NVMe SSDs, and 40GbE network interconnection. PetrelKube is managed with Rancher (https://rancher.com), providing secure authentication and identity management as well as a user-friendly mechanism to deploy and manage services in the cluster. We chose Kubernetes as the basis for this service as it provides a proven flexible and reliable platform to deploy services and address scalable computing needs (such as data analytics).

2.4 Petrel v3 Upgrade

Petrel is currently being upgraded to a Ceph-based storage model that will expand the amount of storage available (to more than 3PB), enable storage to be directly mounted by containers in PetrelKube, and to provide enforceable user allocations.

Petrel v3 hardware will be based on 32 Object Storage Daemon (OSD) servers, each with 12 12TB Seagate EXOS X12 SATA hard drives, 1 Intel Optane P4800X 375GB NVMe SSD (used for Ceph Bluestore WAL/DB), Intel Skylake Xeon-D 2146NT 8c 2.3Ghz, 64GB RAM, and 4×10GbE. The 384 12TB drives will deliver 3.2PB of usable storage using 8+2p erasure coding for data and 3-way data replication. The system will also include 3 Ceph monitoring hosts to manage cluster membership and provide monitoring functions and 2 Ceph Metadata servers to provide CephFS functionality. Petrel v3 will run community releases of Ceph Mimic (13.2.x) on RHEL7.

Petrel v3 will use the same 8 node DTN infrastructure as used in Petrel v2. The Ceph public (Ceph client network) network consists of a pair of Dell S4148T-ON (48x10GbE 10Gbase-T, 4×100GbE QSFP28 ports) and a Dell Z9100-ON (32x 100GbE QSFP port). Nodes are connected via a pair of 10GbE ports. The DTNs are directly connected to the public network via 1x100GbE each. Four 100GbE ports are connected to internal core switches for efficient connection to other resources (e.g., Petrelkube). The Ceph cluster network (for backend Ceph OSD intercommunication only) network consists of a pair of Dell S4148F-ON (48×10GbE SFP+, 4×100GbE QSFP28 ports) and a Dell Z9100-ON (32x 100GbE QSFP port). The OSD servers are connected via a pair of 10GbE ports. The Metadata and monitor nodes do not connect to the cluster network, nor do the DTNs or external resources.

3 EXAMPLE USER WORKFLOWS

Any scientist with an Argonne affiliation can request a project allocation on Petrel. Upon approval, they have access to 100 TB which they can access via Globus web interface and REST APIs, either directly or by using associated Python software development kits (SDKs). They can transfer data to and from Petrel, create and delete directories and shared endpoints, and grant other users rights to manage, read, and/or write the space.

We provide quantitative data on usage in Section 4. Here we review a few representation examples of specific applications that make use or, or that were developed specifically for, Petrel.

3.1 Light Source Data Distribution

The Advanced Photon Source (APS) at Argonne National Laboratory, like many experimental facilities worldwide, serves thousands of researchers every year, most of whom visit just for a few days to collect data and then return to their home institution. In the past, data produced during an experiment were invariably carried back on physical media. However, as data sizes have grown and experiments have become more collaborative, that approach has become less effective. Data transfer via network is preferred; the challenge is to integrate data transfer into the experimental workflow of the facility in a way that is automated, secure, reliable, and scalable.

The DMagic system (dmagic.readthedocs.io) used Petrel to do just that. DMagic integrates with APS administrative and facility systems to deliver data to experimental users. Before the experiment begins, it uses a Globus API to create a shared endpoint on Petrel, retrieves from the APS scheduling system the list of approved users for the experiment, and uses further Globus API calls to add permissions for those users to the shared endpoint. It subsequently monitors the experiment data directory at the APS experimental facility and copies new files automatically to that shared endpoint, from which they can be retrieved by any approved user.

DMagic is now used by several APS beamlines to track study metadata and provenance, and to share raw and analyzed data with local and remote collaborators. For example, a microtomography beamline currently collects 20–80 TB/month of raw data and expects to scale to about 100–200 TB/month in the near future. Petrel is being used in workflows that involve streaming analysis of data as it is collected [6] with the TomoPy toolkit [15], and for experiments in rule-based data collection and processing [12].

3.2 Cosmology:

Scientists at ANL have used Petrel to create a portal (cosmology.alcf.anl.gov) for sharing large gravity-only cosmological simulations generated by the Hardware/Hybrid Accelerated Cosmology Code (HACC) code [?]. The portal offers access to some of the largest cosmological simulations ever conducted such as the Outer Rim simulations that cover a volume of $(4225\ Mpc)^3$ and evolves more than 1 trillion particles. Building on the MRDP pattern and Petrel as a general data management service, the Python-based portal allows other researchers to discover and download huge amounts of data. The portal includes a Javascript-based search model, allowing discovery by model, redshift, datatype, and many other cosmological parameters. Researchers may then select the datasets of interest, and start a Globus transfer to a selected destination endpoint.

3.3 Materials Science

Scientists from Argonne's Materials Science Division gather experimental data from APS beamlines. Raw data volumes range from 60–100 TB/month. They use Petrel and Globus functionality to share, bundle, and publish datasets [?]. They also integrate Petrel into workflows that make datasets accessible for remote visualization from Python clients [?].

After data are gathered, the scientists often want to share subsets of raw data, derived datasets, and analysis results with collaborators, track metadata associated with the data, and track data provenance. Eventually, these scientists may want to make their datasets publicly and persistently available via publication functionality [10], fully bundled with the associated metadata and associated with a persistent identifier to aide search, discovery, and data citability.

3.4 Materials Data Facility

The Materials Data Facility (MDF) [7] provides the materials science community with data services for simplified publication, discovery, and reuse of materials-related datasets. Its data publication service allows researchers to publish datasets stored on distributed resources (i.e., any Globus endpoint), mint permanent unique identifiers (e.g., DOI or Handle), and operate data publication workflows through a web interface or API [10]. The MDF data discovery service provides a simple query API along with Python tools to help users find and access full datasets published in MDF as well as discover links to a host of relevant data indexed from a variety of external efforts (117 sources indexed to date).

MDF leverages the Petrel service to: 1) share collaborative spaces with research teams across the country for use in collecting and organizing project data prior to final publication and sharing with the public; 2) stage datasets for indexing into the MDF discovery service (currently ~60 TB of data); and 3) store a raw copy of the extracted JSON-formatted metadata for index persistence.

In the future, the MDF team is interested in attaching compute resources to Petrel in order to streamline operations on the stored datasets. These operations may include indexing material-specific file contents, extracting general metadata, performing analyzing and modeling tasks, visualizing dataset components, submitting jobs to other ALCF resources, or interacting with the data via applications such as Jupyter [?].

3.5 Scalable inference:

PetrelKube enables a variety of tools and services to perform analytics on Petrel data. One example of this is in the Data and Learning Hub for Science (DLHub) [?]—a Machine Learning (ML) publication and serving infrastructure, designed to make ML models more accessible. ML practitioners can deposit models into DLHub and have them securely served on PetrelKube. The high-performance, low-latency data access between Petrel and PetrelKube makes it an ideal platform for staging large datasets into the models. Users can reference datasets stored on Petrel as inputs to DLHub models and have them securely used by the model containers for inference DLHub uses Parsl [?] to manage the scalable execution of deployed models on PetrelKube.

3.6 Other applications

Petrel file system data and metadata have been used in studies of rule-based data management [12] and automated type inference and metadata extraction [?], and for data transfer modeling and optimization studies [?]. In each case, the ability to write programs that stage data on high-speed storage has proved useful.

The DOE's Energy Exascale Earth System Model project (formerly ACME) [?] uses Petrel to share large climate simulation data, including those published via the Earth System Grid [30].

4 USAGE DATA

Petrel has operated in its current configuration since July 14, 2014. (A first version, constructed with repurposed hardware from the Intrepid supercomputer, used 32 1 Gbps-connected DTNs and DDN S2A9900 storage systems.) During that time, Petrel has been used to perform 119,011 transfers (66,931 outbound, 52,648 inbound), comprising 9.06 PB (3.72 PB outbound, 5.90 PB inbound) and 517M files (142M outbound, 384M inbound). (Numbers do not add up perfectly due to a few hundred transfers from Petrel to Petrel.) Usage is distributed across the Globus web interface (21,558 transfers), Globus Command Line Interface (22,487 transfers), and REST APIs (74,966 transfers). A total of 1120 unique endpoints have communicated with Petrel, 434 as sources and 942 as destinations. Figure 2 shows that these endpoints are located primarily in the US and Europe, but also encompass Asia and Australia. (Many endpoints located within Argonne National Laboratory are not visible on the map as their distance from Petrel is too small.)

Figure **??** provides some perspectives on Petrel usage and performance. Each point in this graph represents a single transfer, often encompassing multiple files but managed as a unit. The x-axis represents the great circle distance between Petrel and the remote source or destination and the y-axis, with a log scale, the number of bytes involved in the transfer. From left to right, we see many transfers within the US, a considerable number to Europe (the great circle distance from Chicago to London is 6,300 km), and significant numbers even to yet more distant locations. Looking at sizes, we see many transfers of more than a terabyte and one close to a petabyte, but the majority quite a bit smaller. Transfer rates also vary considerably. They are in general lower for smaller and more distant transfers; the highest observed was 40 Gbps (5 GB/s) disk-to-disk from the University of Chicago's Research Computing Center. Given that Petrel is itself optimized for high-speed transfers, we can expect that, as observed in other studies [? ? ? ? ?], the rate for any particular transfer will depend largely on the characteristics of the transfer and the remote endpoint, and on the volume of contending transfers.

Figure 3 shows data size vs transfer rate for transfers that originate or end at Petrel. The graph shows particular emphasis on

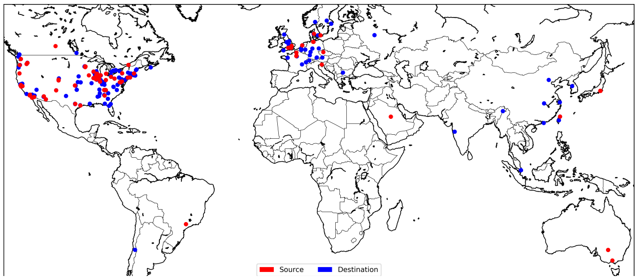

Figure 2: The 1098 (of 1120 total) Petrel source and destination endpoints for which geolocation data are available.

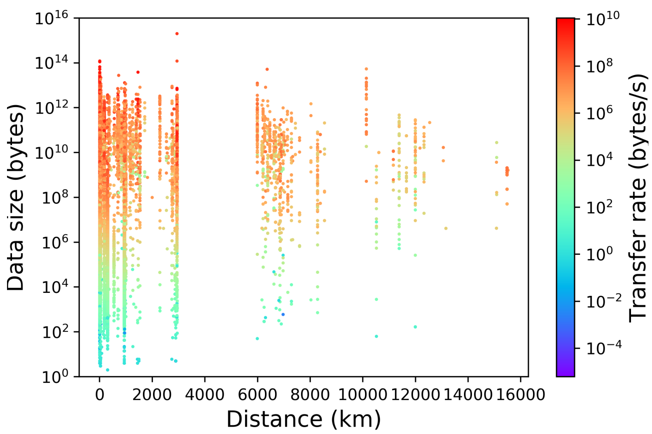

Figure 3: The 79,415 transfers with Petrel as source or destination for which geolocation data are available. Each point represents a single transfer, and gives size vs. great circle distance between Petrel and remote source or destination, with transfer rate color coded.

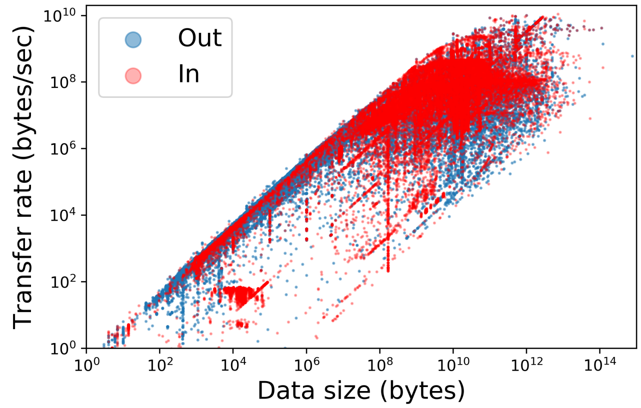

Figure 4: Transfer rate vs. data size. Each point represents a single transfer request, which may involve many files. Incoming and outgoing transfers are distinguished.

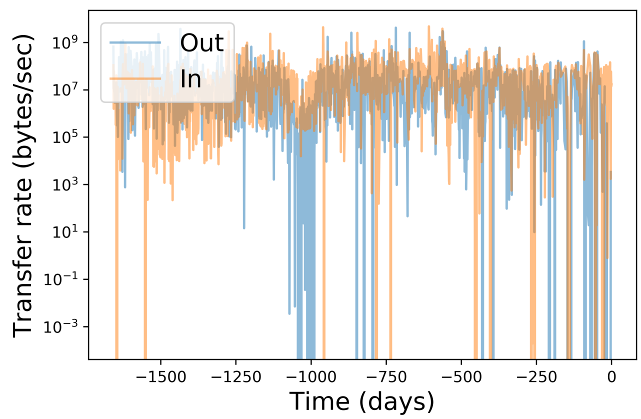

Figure 5: Input/output transfer volumes per day. The numbers on the x axis represent days prior to February 2, 2019.

large-scale and high performance data transfers coming into and out of Petrel. Outgoing transfers are in general more variable with respect to size and transfer rate as there are significantly more outgoing transfers and significantly more destination endpoints.

Petrel has proved to be highly reliable: only one hard drive has failed since it was put into production, in July 2017. (One other drive failed during early burn-in.)

5 RELATED WORK

Many systems have been developed to enable access to scientific data, from GenBank [5] and the Earth System Grid [30] to Dataverse [21] and the Structural Biology Grid [21]. However, few follow the MRDP design pattern [8] in which data are located within a Science DMZ for high-speed access. Nor do the various portals [22], science gateways [19, 29], hubs [18, 20], and cloud-hosted systems [2] that have been developed to enable on-demand access to science software. But these systems could all be adapted to make use of Petrel-like capabilities.

A growing number of research computing centers and associated facilities operate Science DMZs with associated DTNs for remote access to large storage [13]. However, the storage itself is often accessible only by users with facility accounts and is optimized for access by high-performance computing systems, with efficient DTN access a secondary consideration. Petrel, in contrast, is optimized for high-speed network access and enables access by anyone.

Another approach to providing data services for science is to deploy a distributed file system across collaborating systems and sites. The Andrew File System [17] has been used for this purpose. Indiana University's Data Capacitor uses the Lustre parallel distributed file system [25], a technology that was also explored on the TeraGrid [24]. MeDiCI leverages file system mechanisms to enable data caching on a metropolitan scale [1]. File system features such as caching can be useful, but require tighter integration across participating systems, for example at the level of accounts.

Public and private cloud computing systems can also be used for online analysis of large datasets [14, 16]. Szalay advocates for specialized hardware platforms for online analysis of large simulation datasets [26]. Such approaches are largely orthogonal to the software architecture used here.

Petrel and the MRDP design pattern emphasize the placement of storage outside the HPC center and programmatic control of data placement, access, and movement. Logistical networking [3] takes this idea much further, embedding storage throughout the network to be managed by specialized protocols [4].

6 CONCLUSIONS

We have described the realization of a high-speed data service. Our experience operating this service over the past four years suggests that this service is indeed useful to many people. As expected, researchers have used it to distribute data produced at Argonne in computational and experimental studies and to stage data being brought to the ALCF for analysis. Unexpected but pleasing is the variety of ways of which researchers have leveraged Petrel's API access to integrate it into application workflows.

Petrel's proximity to ALCF resources makes it easy for application workflows to transfer data to high-performance computers for analysis. However, some groups would like yet more tightly coupled computing to allow more efficient responses to, for example, user requests for data subsets or analyses. We thus plan to enhance Petrel to support data analysis as well as access. Care must be taken when so doing to address security concerns. Access to ALCF production resources requires an approval process and issuance of a CRYPTOCard hardware access token. Petrel, in contrast, is treated more like a web server, allowing access by anyone who can authenticate with an approved credential. One promising approach is to allow general access only for predefined analysis requests, while requiring two-factor authentication for user-specified analyses.

ACKNOWLEDGMENTS

This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. This work was also supported in part by NSF grant ACI-1148484. We gratefully acknowledge the computing resources provided and operated by the Joint Laboratory for System Evaluation (JLSE) at Argonne National Laboratory.

REFERENCES

[1] David Abramson, J. Carroll, C. Jin, and M. Mallon. 2017. A Metropolitan Area Infrastructure for Data Intensive Science. In *13th IEEE eScience Conference*.
[2] Yadu N. Babuji, Kyle Chard, A. Gerow, and Eamon Duede. 2016. Cloud Kotta: Enabling secure and scalable data analytics in the cloud. In *IEEE International Conference on Big Data*. 302–310. https://doi.org/10.1109/BigData.2016.7840616
[3] Micah Beck, Terry Moore, Jim Plank, and Martin Swany. 2000. Logistical networking. In *Active Middleware Services*. Springer, 141–154.
[4] Micah Beck, Terry Moore, and James S Plank. 2002. An end-to-end approach to globally scalable network storage. In *ACM SIGCOMM Computer Communication Review*, Vol. 32. ACM, 339–346.
[5] Dennis A Benson, Mark Cavanaugh, Karen Clark, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, and Eric W Sayers. 2012. GenBank. *Nucleic Acids Research* 41, D1 (2012), D36–D42.
[6] Tekin Bicer, Doğa Gürsoy, Rajkumar Kettimuthu, Francesco De Carlo, and Ian T Foster. 2016. Optimization of tomographic reconstruction workflows on geographically distributed resources. *Journal of Synchrotron Radiation* 23, 4 (2016), 997–1005.
[7] Ben Blaiszik, Kyle Chard, Jim Pruyne, Rachana Ananthakrishnan, Steven Tuecke, and Ian Foster. 2016. The Materials Data Facility: Data Services to Advance Materials Science Research. *Journal of Materials* 68, 8 (2016), 2045–2052.
[8] Kyle Chard, Eli Dart, Ian Foster, David Shifflett, Steven Tuecke, and Jason Williams. 2017. The Modern Research Data Portal: A Design Pattern for Networked, Data-Intensive Science. *PeerJ Preprints* (2017). docs.globus.org/mrdp.
[9] Kyle Chard, Mattias Lidman, Brendan McCollam, Josh Bryan, Rachana Ananthakrishnan, Steven Tuecke, and Ian Foster. 2016. Globus Nexus: A Platform-as-a-Service provider of research identity, profile, and group management. *Future Generation Computer Systems* 56 (2016), 571 – 583. https://doi.org/10.1016/j.future.2015.09.006
[10] Kyle Chard, Jim Pruyne, Ben Blaiszik, Rachana Ananthakrishnan, Steven Tuecke, and Ian Foster. 2015. Globus data publication as a service: Lowering barriers to reproducible science. In *11th International Conference on e-Science*. IEEE, 401–410.
[11] Kyle Chard, Steve Tuecke, Ian Foster, Bryce Allen, Rachana Ananthakrishnan, Joe Bester, Ben Blaiszik, Vytas Cuplinskas, Raj Kettimuthu, Jack Kordas, Lukasz Lacinski, Mattias Lidman, Mike Link, Stu Martin, Brendan McCollam, Karl Pickett, Dan Powers, Jim Pruyne, Brigitte Raumann, Gigi Rohder, Stephen Rosen, Dave Shifflett, Teresa Sutton, Vas Vasiliadis, and Jason Williams. 2016. Globus: Recent Enhancements and Future Plans. In *5th Annual XSEDE Conference*.
[12] Ryan Chard, Kyle Chard, Jason Alt, Dilworth Y Parkinson, Steve Tuecke, and Ian Foster. 2017. Ripple: Home automation for research data management. In *37th International Conference on Distributed Computing Systems Workshops*. IEEE, 389–394.
[13] Eli Dart, Lauren Rotman, Brian Tierney, Mary Hester, and Jason Zurawski. 2013. The Science DMZ: A Network Design Pattern for Data-intensive Science. In *International Conference on High Performance Computing, Networking, Storage and Analysis (SC '13)*. ACM, New York, NY, USA, Article 85, 10 pages. https://doi.org/10.1145/2503210.2503245
[14] Ian Foster and Dennis Gannon. 2017. *Cloud Computing for Science and Engineering*. MIT Press. https://cloud4scieng.org.
[15] Doğa Gürsoy, Francesco De Carlo, Xianghui Xiao, and Chris Jacobsen. 2014. TomoPy: A framework for the analysis of synchrotron tomographic data. *Journal of Synchrotron Radiation* 21, 5 (2014), 1188–1193.
[16] Allison P Heath, Matthew Greenway, Raymond Powell, Jonathan Spring, Rafael Suarez, David Hanley, Chai Bandlamudi, Megan E McNerney, Kevin P White, and Robert L Grossman. 2014. Bionimbus: A cloud for managing, analyzing and sharing large genomics datasets. *Journal of the American Medical Informatics Association* 21, 6 (2014), 969–975.
[17] John H Howard, Michael L Kazar, Sherri G Menees, David A Nichols, Mahadev Satyanarayanan, Robert N Sidebotham, and Michael J West. 1988. Scale and performance in a distributed file system. *ACM Transactions on Computer Systems* 6, 1 (1988), 51–81.
[18] Gerhard Klimeck, Michael McLennan, Sean P Brophy, George B Adams III, and Mark S Lundstrom. 2008. nanohub.org: Advancing education and research in nanotechnology. *Computing in Science & Engineering* 10, 5 (2008), 17–23.
[19] Katherine A Lawrence, Michael Zentner, Nancy Wilkins-Diehr, Julie A Wernert, Marlon Pierce, Suresh Marru, and Scott Michael. 2015. Science gateways today and tomorrow: Positive perspectives of nearly 5000 members of the research community. *Concurrency and Computation: Practice and Experience* 27, 16 (2015), 4252–4268.
[20] Michael McLennan and Rick Kennell. 2010. HUBzero: A platform for dissemination and collaboration in computational science and engineering. *Computing in Science & Engineering* 12, 2 (2010).
[21] Peter A. Meyer, Stephanie Socias, Jason Key, Elizabeth Ransey, Emily C. Tjon, Alejandro Buschiazzo, Ming Lei, Chris Botka, James Withrow, David Neau, Kanagalaghatta Rajashankar, Karen S. Anderson, Richard H. Baxter, Stephen C. Blacklow, Titus J. Boggon, Alexandre M. J. J. Bonvin, Dominika Borek, Tom J. Brett, Amedeo Caflisch, Chung-I Chang, Walter J. Chazin, Kevin D. Corbett, Michael S. Cosgrove, Sean Crosson, Sirano Dhe-Paganon, Enrico Di Cera, Catherine L. Drennan, Michael J. Eck, Brandt F. Eichman, Qing R. Fan, Adrian R. Ferré-D'Amaré, J. Christopher Fromme, K. Christopher Garcia, Rachelle Gaudet, Peng Gong, Stephen C. Harrison, Ekaterina E. Heldwein, Zongchao Jia, Robert J. Keenan, Andrew C. Kruse, Marc Kvansakul, Jason S. McLellan, Yorgo Modis, Yunsun Nam, Zbyszek Otwinowski, Emil F. Pai, Pedro José Barbosa Pereira, Carlo Petosa, C. S. Raman, Tom A. Rapoport, Antonina Roll-Mecak, Michael K. Rosen, Gabby Rudenko, Joseph Schlessinger, Thomas U. Schwartz, Yousif Shamoo, Holger Sondermann, Yizhi J. Tao, Niraj H. Tolia, Oleg V. Tsodikov, Kenneth D. Westover, Hao Wu, Ian Foster, James S. Fraser, Filipe R. N C. Maia, Tamir Gonen, Tom Kirchhausen, Kay Diederichs, Mercè Crosas, and Piotr Sliz. 2016. Data publication with the Structural Biology Data Grid supports live analysis. *Nature Communications* 7 (2016).
[22] Michael Russell, Gabrielle Allen, Greg Daues, Ian Foster, Edward Seidel, Jason Novotny, John Shalf, and Gregor Von Laszewski. 2001. The Astrophysics Simulation Collaboratory: A science portal enabling community software development. In *10th IEEE International Symposium on High Performance Distributed Computing*. 207–215.
[23] Frank B Schmuck and Roger L Haskin. 2002. GPFS: A Shared-Disk File System for Large Computing Clusters. In *USENIX Conference on File and Storage Technologies*, Vol. 2.

[24] Stephen C Simms, Gregory G Pike, and Douglas Balog. 2007. Wide area filesystem performance using Lustre on the TeraGrid. In *TeraGrid Conference*.

[25] Stephen C Simms, Gregory G Pike, Scott Teige, Bret Hammond, Yu Ma, Larry L Simms, C Westneat, and Douglas A Balog. 2007. Empowering distributed workflow with the Data Capacitor: Maximizing Lustre performance across the wide area network. In *Workshop on Service-oriented Computing Performance: Aspects, Issues, and Approaches*. ACM, 53–58.

[26] Alexander S Szalay. 2014. From simulations to interactive numerical laboratories. In *Winter Simulation Conference*. IEEE Press, 875–886.

[27] S. Tuecke, R. Ananthakrishnan, K. Chard, M. Lidman, B. McCollam, S. Rosen, and I. Foster. 2016. Globus auth: A research identity and access management platform. In *IEEE 12th International Conference on e-Science (e-Science)*. 203–212. https://doi.org/10.1109/eScience.2016.7870901

[28] Thomas D. Uram and Michael E. Papka. 2016. Expanding the Scope of High-Performance Computing Facilities. *Computing in Science and Engineering* 18, 3 (May 2016), 84–87. https://doi.org/10.1109/MCSE.2016.53

[29] Nancy Wilkins-Diehr, Dennis Gannon, Gerhard Klimeck, Scott Oster, and Sudhakar Pamidighantam. 2008. TeraGrid science gateways and their impact on science. *Computer* 41, 11 (2008).

[30] Dean N. Williams, R. Drach, Rachana Ananthakrishnan, Iand T. Foster, Dan Fraser, Frank Siebenlist, David E. Bernholdt, M. Chen, J. Schwidder, S. Bharathi, Ann L. Chervenak, Rob Schuler, Mei Su, D. Brown, Luca Cinquini, Peter Fox, Jose Garcia, Don E. Middleton, W. Gary Strand, N. Wilhelmi, S. Hankin, R. Schweitzer, P. Jones, Arie Shoshani, and Alex Sim. 2009. The Earth System Grid: Enabling Access to Multimodel Climate Simulation Data. *Bulletin of the American Meteorological Society* 90, 2 (2009), 195–205. https://doi.org/10.1175/2008BAMS2459.1