



PDACS: A Portal for Data Analysis Services for Cosmological Simulations

Ravi Madduri | Argonne National Laboratory

Alex Rodriguez | University of Chicago

Thomas Uram and Katrin Heitmann | Argonne National Laboratory

Tanu Malik | University of Chicago

Saba Sehrish | Fermi National Accelerator Laboratory

Ryan Chard | Victoria University of Wellington

Shreyas Cholia | Lawrence Berkeley National Laboratory

Marc Paterno and Jim Kowalkowski | Fermi National Accelerator Laboratory

Salman Habib | Argonne National Laboratory

A Web-based analysis portal provides access both to large simulations and parallel analysis tools and to opportunities to access, transfer, manipulate, search, and record simulation data. The system allows for cross-layer provenance tracking and implementing a transparent method for sharing workflow specifications, offering a convenient mechanism for checking reproducibility.

Large-scale simulations of structure formation in the universe performed on leadership-class supercomputing systems are a key tool of discovery in modern cosmology. The simulations provide essential elements of the theoretical and modeling counterpart to deep sky survey observations that probe a large fraction of the available sky.¹ The combination of theory and observation has led to a remarkably successful cosmological model that has not only revolutionized our knowledge of the universe's composition and evolution but also provides compelling pointers—such as dark matter and dark energy—to discoveries in fundamental physics.

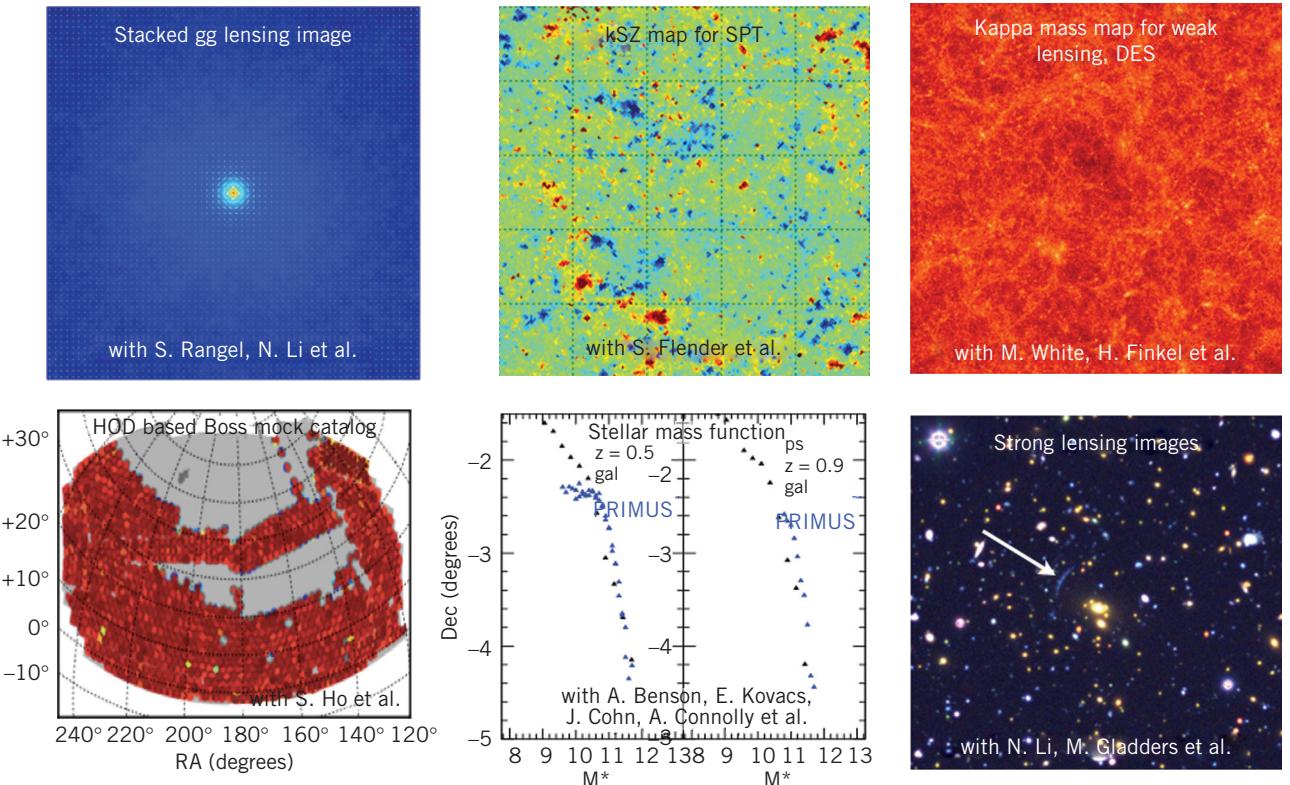


Figure 1. Examples of different science results extracted from cosmological N -body simulations. Top row (left to right): a stacked galaxy-galaxy lensing result as it would be measured by DES (Dark Energy Survey) and LSST (Large Synoptic Survey Telescope), a map showing the kinematic Sunyaev-Zeldovich effect as it would be measured by SPT (South Pole Telescope), and a weak lensing map as it would be measured by DES and LSST. Bottom row (left to right): a galaxy map showing BOSS (Baryon Oscillation Spectroscopic Survey) or DESI (Dark Energy Spectroscopic Instrument)-like structures, the stellar mass function as it would be measured by LSST, and a strong lensing image from a cluster survey.

The simulation results derive from a major investment in scarce supercomputer resources and specialized high-performance computing (HPC) codes. Only a small number of these codes exist worldwide, but their outputs are crucial for many researchers as well as large project teams working on the analysis of major ongoing and near-future optical sky surveys, including BOSS (Baryon Oscillation Spectroscopic Survey), DES (Dark Energy Survey), DESI (Dark Energy Spectroscopic Instrument), LSST (Large Synoptic Survey Telescope), microwave surveys such as ACT (Atacama Cosmology Telescope) and SPT (South Pole Telescope), and radio surveys such as the SKA (Square Kilometer Array). Consequently, broadly sharing simulation outputs² and providing efficient analysis tools that can handle large datasets are matters of ever-increasing importance. The results from a single simulation can be used in many different ways, and researchers can perform many different downstream

science analyses and processing sequences by using them. Figure 1 shows a set of examples extracted from large-scale structure simulations.

Simulation datasets today can have sizes spanning hundreds of terabytes up to several petabytes of data, limited only by the size of available file systems. The analysis of such large outputs requires efficient parallel analysis tools, which run on large supercomputers or analysis clusters and typically proceed along two well-defined paths: offline and inline (or as the latter is often termed, *in situ*). Offline analysis refers to interrogating, transforming, and visualizing simulation data accessible via file systems and databases. In contrast, *in situ* analysis is the carrying out of analogous tasks on the host (typically, HPC) system while the simulation is underway. A thoughtfully designed global analysis strategy must optimally combine offline and *in situ* analysis tasks, both of which can require significant familiarity with parallel processing issues. Many

researchers in cosmology aren't trained in writing parallel code (nor are they interested in becoming so) or lack individual accounts on large-scale computing resources. Providing easy access to simulation outputs, along with the means to analyze the data and extract as much new science as possible, is thus extremely desirable. Finally, to understand and verify some of the results obtained from a simulation by a team that didn't perform the original, potentially complex, analysis, it's important to have easy ways to share analysis workflows.

For these reasons, we developed PDACS (Portal for Data Analysis Services for Cosmological Simulations), a Web-based analysis portal for offline analysis that provides access to large simulations and analysis tools, along with opportunities to access, transfer, manipulate, search, and record simulation data. PDACS lets users wrap analysis tools written in a large number of languages, including both structured and database programming languages, and makes them available within its workflow system, providing a powerful way to perform multi-level or multistep analyses. The system also allows for cross-layer provenance tracking across analysis tools, thereby implementing a transparent sharing method, as well as a convenient mechanism for checking reproducibility. Users are able to submit their own tools to the system and share them with the community.

The current PDACS version supplies a set of simulations from the extended Coyote Universe suite,³ covering 37 different cosmological models that span different volumes and resolutions. It also provides access to a set of parallel and serial analysis tools to investigate the simulations, as well as a set of stand-alone tools the user can employ to conduct database-enabled analyses of simulation outputs or compare results against other approximate methods. The results can be interactively viewed and plotted within the workflow through a self-contained analysis engine that allows exploration of a large set of cosmological simulations. PDACS currently runs at the National Energy Research Scientific Computing Center (NERSC) and on Magellan, a cloud resource located at Argonne National Laboratory. Deployment at Oak Ridge National Laboratory is currently underway.

Implementation and Deployment

PDACS is implemented using the Galaxy platform.⁴ Originally developed for biomedical applications, Galaxy provides a Web-based interface for creating, executing, sharing, and reusing workflows

composed of arbitrary applications, tools, and scripts. Galaxy records users' analysis steps, and at the completion of a task, lets them save the constituent steps as a workflow to re-execute later or to share with others. Thousands of biomedical researchers have used Galaxy; it's currently actively used by hundreds of developers around the world, including members of the PDACS team.

When we performed a requirements analysis for PDACS, Galaxy provided the required core capabilities of interactive execution on HPC resources, workflow creation, sharing, and provenance. Its extensible plug-in architecture lets users customize data types and tools for any domain. Before being usable in Galaxy, tools must be wrapped with XML files that describe their input and output data types, executable location, and invocation arguments. Galaxy uses this information to present customized user interfaces with specific input parameters, linking the output of one tool to the input of another. Galaxy enforces type checking by providing visual feedback to users when connecting tools that have incompatible inputs and outputs. It also provides a tool repository (called a tool shed) to store and share tool wrappers, data types, and exported workflows. A global tool shed operated by the Galaxy team provides thousands of freely available (biomedical) tools that can be installed on any Galaxy deployment. Currently, PDACS workflows use a visual directed acyclic graph interface for ease of use, but in the future, we aim to include a notebook-style interface as well (such as Jupyter).

In previous work, we integrated Galaxy with Globus services to create a general-purpose platform for scientific data management and analysis called Globus Galaxies. We successfully used Globus Galaxies to create science gateways for analyzing cardiovascular, crop-modeling, genomic, and imaging data. The platform is built on the Galaxy workflow system for flexibly creating and executing scientific workflows; Globus Transfer⁵ for easy, reliable, and secure access, sharing, and transfer of large amounts of data; and Globus Nexus⁶ for identity management and group-based access control. For Galaxy to be easily deployed over various computational infrastructures, we developed job runners, which provide an abstraction from the complexity of using the supporting infrastructure to enable the execution of workloads over different HPC platforms. In PDACS, we customized the Globus Galaxies platform for cosmology use cases by adding data types for cosmology datasets, creating Galaxy wrappers for various cosmology applications,

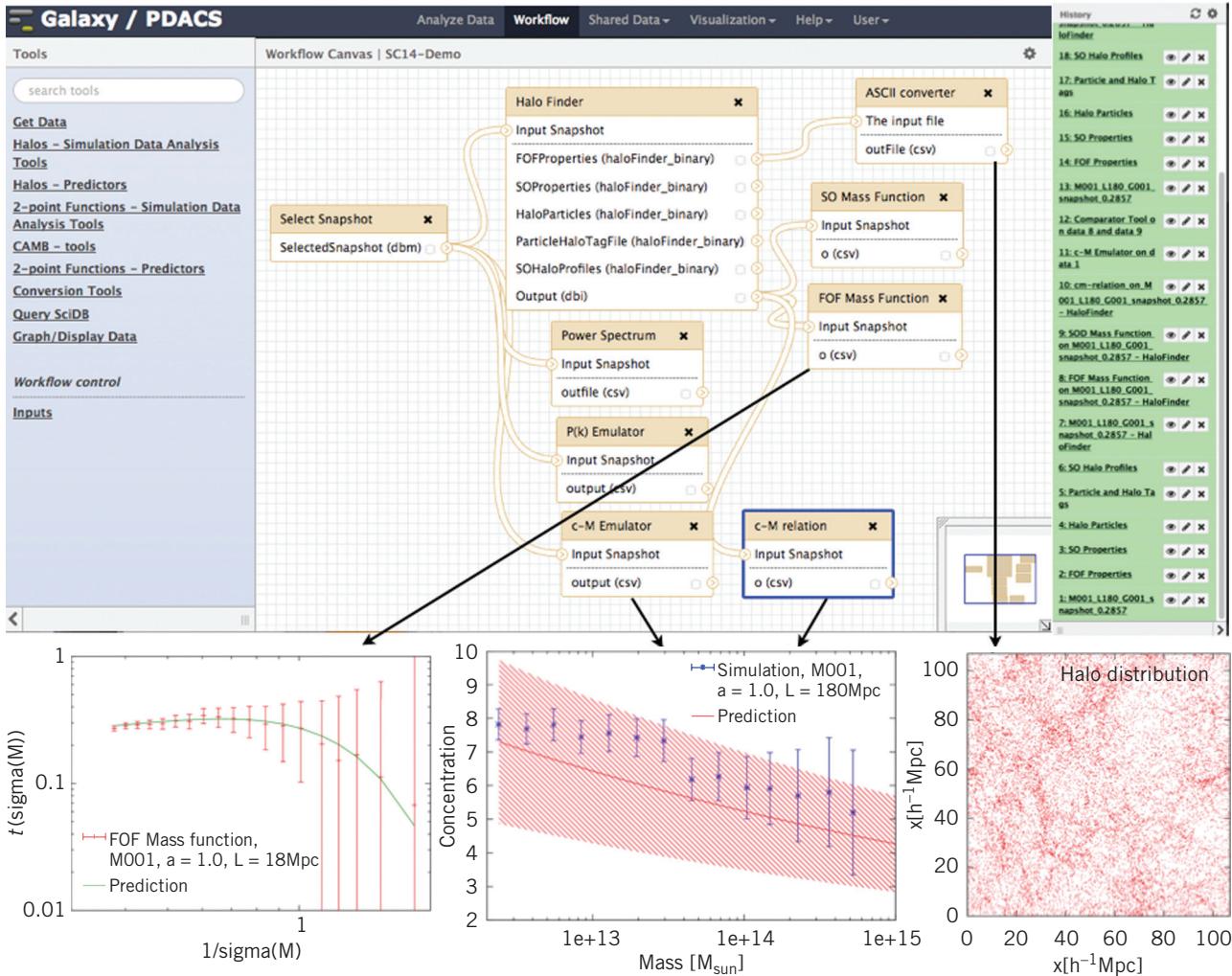


Figure 2. Screenshot of a PDACS workflow. The left (blue) panel shows the toolbar, the middle section the workflow canvas, and the right (green) panel, job status (green indicates successful completion). Here, a whole suite of analysis tasks was performed on top of the simulation, and predictions for various quantities were generated. The three lower panels—also generated in PDACS—show images from the outputs.

and integrating them with NERSC and Magellan identity management and execution management services.

Analyzing and Sharing Cosmological Data Using PDACS

We developed PDACS with two major goals in mind: sharing data and analysis tools, and providing an easy-to-use interface for analysis tasks with large-scale computing resources. The current PDACS version provides access to a set of almost 150 simulations, covering 37 different cosmological models. At the start of an analysis task, users can choose one or more simulations or stand-alone tools. They can also upload and share their own datasets with the

PDACS community. In Figure 2, “Get data” shows how you choose a simulation set or import data; stand-alone tools are shown as “Predictors.”

PDACS provides a set of frequently used cosmology tools: friends-of-friends (FOF) and spherical overdensity (SO) halo finders (tools to locate local mass overdensities, or clumps of neighboring particles, in the simulation results); halo radial profile measurements, including concentration (a parameter that characterizes a halo profile); binning routines to calculate the halo mass distribution (the mass function) from the halo finder outputs; a concentration-mass relation emulator that predicts the correlation between concentration and halo mass; a density power spectrum emulator; and measurement

tools for two-point statistics (the fluctuation power spectrum and correlation function). All these tools are shown in the left panel of Figure 2. Some of them are written in C++/MPI, some in Python, and some in C. The PDACS philosophy is to support any language for the analysis tools as long as the codes run flawlessly on the analysis hardware. Computationally intensive tools, such as the halo finder, are automatically submitted as batch jobs to be run on supercomputing resources, while less intensive tools, such as the ones for prediction, are executed on the same node that manages the Galaxy instance.

As installed at NERSC, PDACS leverages the NERSC Web Toolkit (NEWT) for authenticating users and dispatching jobs.⁷ Authenticating users with NERSC credentials associates jobs and workflows dispatched through PDACS with a NERSC user, allowing resource accountability and the enforcement of compute and data restrictions. The PDACS platform operating on the Magellan cloud authenticates users with single sign-on through Shibboleth credentials. The Magellan cloud resource is provisioned via the OpenStack platform, and parallel analysis jobs submitted to the PDACS instance at Magellan run on a virtual cluster constructed on demand. The PDACS job specification includes the number of nodes and cores per node that should be used to run the job. The PDACS job runner on Magellan communicates with OpenStack to instantiate the required virtual machine instances, using a pre-built system image that includes the analysis tools and supporting libraries.

The instances are configured for access to the data volumes and with the requisite networking to behave as a message passing interface (MPI) cluster. Once the parallel job has successfully executed on this virtual cluster, the virtual machine instances are destroyed and their resources are returned to the pool. As demand for analysis services on Magellan increases, this model will be adapted to support an elastic virtual cluster. In this way, a collection of virtual clusters will be readily available to process jobs in the queue, avoiding the startup cost associated with instantiating a cluster per job. The collection of clusters will grow to meet increases in demand and diminish as the need for resources declines. The PDACS model assumes that, for the most part, computation will move to the data. After choosing which simulation to analyze, the user creates a workflow within PDACS either by using the Workflow Canvas shown in Figure 2 or by choosing analysis tasks in a sequential order (not shown in the figure).

In our example, we combined a diverse set of parallel and serial tools; the Halo Finder and Power Spectrum tools are submitted to the cluster's back end (the user can specify the number of cores to be used), while the other tools ($P(k)$ Emulator, c-M emulator, c-M relation, ASCII converter, SO mass function, and FOF mass function) run on the front end. PDACS carries metadata information to ensure that the user only connects analysis tools that can interact with each other. For example, it isn't possible to connect the Halo Finder tool to the Power Spectrum's output because doing so wouldn't represent a viable scientific workflow. Once the workflow is created, it can be saved, run, and shared with different users. The right panel in Figure 2 monitors job status: submission is in gray, running in yellow, successful execution in green, and unsuccessful execution in red. An important feature of the workflow setup is that the user can choose to perform the analysis for the chosen cosmological model in a self-consistent way, meaning that all cosmological parameters are automatically passed to the different analysis tools.

Navigating from the right panel, the user can now explore the analysis results. It's possible to view the output directly (if it generated an ASCII file) by clicking an eye-like icon; the data will then be displayed in the middle panel. The data can also be downloaded or plotted via different graphic interfaces. The three images in Figure 2's lower panel were generated with gnuplot, a commonly used graphics package. In addition, jobs can be rerun with different parameter settings or log files, and error reports can be investigated. The interface is intuitive and gives the user many options to interact with the data. Another more sophisticated visualization option included in PDACS is ParaViewWeb (<http://paraviewweb.kitware.com>), a collection of components designed to allow the use of ParaView's parallel analysis and visualization capabilities from a Web front end (see Figure 3).

Metadata Handling

Galaxy can recognize data introduced into it through a combination of what Galaxy calls the file's data type and format. If Galaxy recognizes the file type, information (metadata) describing the file's contents can be readily extracted and stored in its metadata database. In life sciences use cases, metadata is usually stored in the file being loaded and can be viewed in the Galaxy Web interface. In cosmology, metadata, such as cosmological model, size, seed, Hubble constant value, and so on, is stored in a different file.

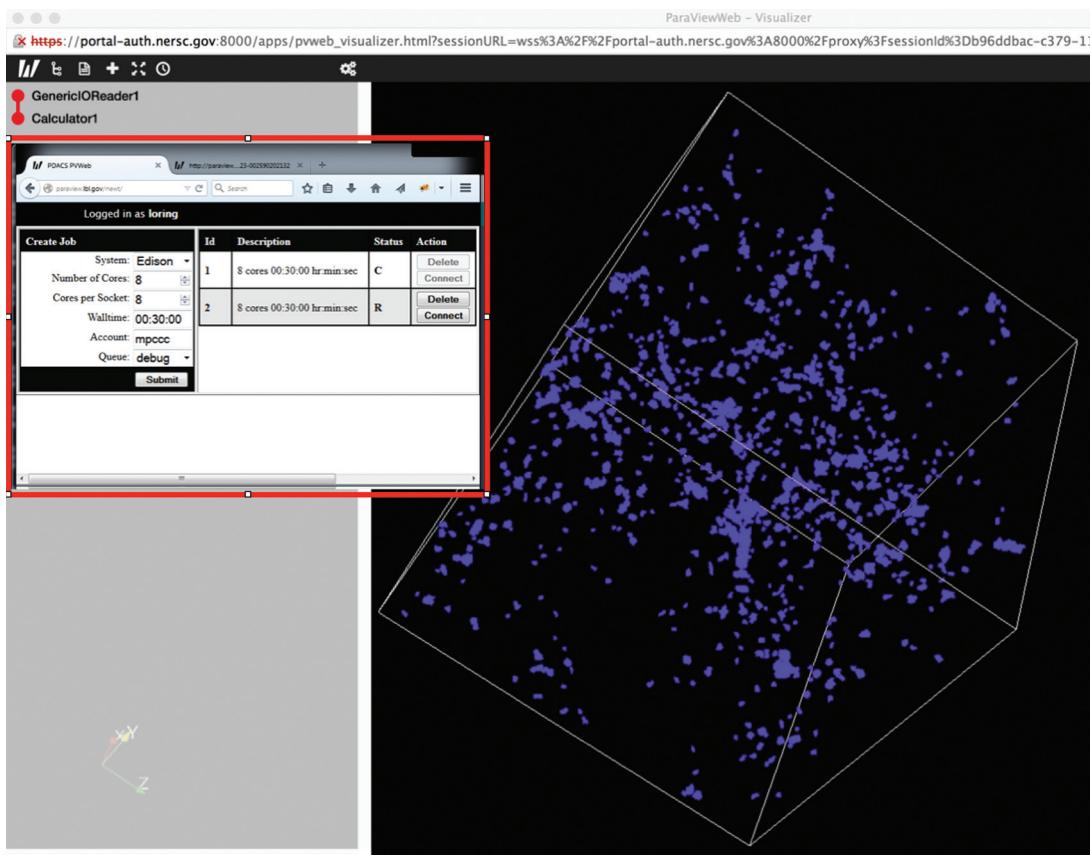


Figure 3. Screenshot showing ParaViewWeb accessed from within PDACS (inset) for a parallel visualization task. This shows a fully interactive 3D visualization carried out from within PDACS.

One way to distinguish two simulation datasets is to look at the metadata file in the simulation data, which is made available with PDACS and its well-defined data hierarchy. In PDACS, we created a simple SQLite data type that can read and store metadata for each dataset that's being loaded. This data type is passed along to various steps in the workflow, so every step can access the original while adding new metadata information as it passes through different steps in the workflow.

In the future, we plan to enhance metadata handling in two ways. First, we'll leverage the built-in Galaxy database with a new table for the cosmology metadata and modify the tools to read from the table. Second, we'll use the Globus catalog service to automatically retrieve index metadata from cosmology datasets and provide an API that can then be used in the Galaxy tool wrappers to pass the information through the workflow steps.

Each tool generates one or more output files, which typically consist of tabular data (recognized by Galaxy as a known data type) that's viewable in the browser. Some tools generate binary data

that can then be converted to ASCII format using a converter tool within PDACS. Galaxy's data types allow for type checking when users build workflows, ensuring that they connect applications in the workflow canvas only when the inputs and outputs of connecting applications are compatible (that is, the first application's output matches the connecting application's input).

Database-Enabled Scientific Workflows

Databases can be powerful inherent components of a scientific workflow, serving as a persistent store for inputs and outputs of analyses for later querying, as well as executing analyses that are database aware. Such use of database technology in a cosmology survey was demonstrated by the Sloan Digital Sky Survey (SDSS), which efficiently used relational database technology to make large amounts of data publicly accessible, find relevant subsets of data (needles in a haystack), and conduct spatial analysis.⁸ SDSS also demonstrated how scientific programs that are part of a workflow, such as cluster finding, can benefit from a modern relational database management

system by using available indexes, parallelizing query execution, and using efficient join algorithms.⁹

In PDACS, our goals are, on the surface, similar to SDSS in that we want databases to be integrated with scientific workflows. The two challenges associated with this are that, first, in Galaxy, the use of database technology is limited (it's used mostly to store metadata and task descriptions), and second, PDACS workflows are much more complex and arbitrary. Unlike SDSS, where the data processing workflow and its component analyses were controlled (limited to range and spatial querying), a generic PDACS use case can involve very diverse workflows. This is due to the presence of several distinct types of analyses, ranging from simple clustering to probabilistic analyses, which can now be integrated on the fly using PDACS. This multiplicity of tools and their potential arbitrary conjunctions through PDACS raises new implications for the use of database technology.

Traditionally, databases are curated and maintained with a given schema. However, workflows generate different datasets, and if persisted in databases, will result in different schemas. Given the volume of workflows that can potentially run through PDACS, setting up every database with a workflow-specific schema can be a significant data management overhead. In PDACS, we simplify this overhead in two ways. First, we wrap the database itself as a PDACS tool so that it can be easily plugged in with the other cosmology tools. This allows for a database to be part of the scientific workflow and visible to other analysis tools for data ingest. Currently, we have data types for relational databases (MySQL and Postgres), and array databases (SciDB).¹⁰ Second, we use data I/O libraries for writing data from analysis tools to scientific files to set up workflow-specific databases. In PDACS, most analysis tools use data I/O libraries such as GenericIO, HDF5, or NetCDF. We use these libraries to determine the exact schema—including the number of files, number of variables, and their data types—of the analysis output.

Consider an analysis tool that finds halos using the simulation datasets, and then outputs a dataset describing the particles and the halos to which they belong, along with a dataset describing the properties of the halo, its position, mass, angular momentum, and so on. Using the Galaxy system metadata and history, we determine how many different kinds of datasets are written by the halo-finder analysis tool, and then, using the tool's GenericIO data I/O library reader and a few sample dataset files that were written, determine the name

and information type of each variable in the dataset. This provides us with all the information to set up a database to ingest the data from this tool. In PDACS, both relational and array databases can be set up. The actual database setup is accomplished by automatically generating corresponding `create table` (for relational) and `create array` (for array) statements, with specific column, dimension, and attribute information, respectively.

We can further simplify the database creation step by providing custom tools that can load the file-based data into the database. These tools read the full dataset, convert the binary formats in which the data was written by the analysis tools into a database-specific format, and then load them in parallel into the database. PDACS's job submission and queuing support provides for efficient and reliable data loading. The loader tool itself combines data I/O readers with existing loaders as available through the databases.

In PDACS, range, aggregate, and probabilistic querying can run on data stored in databases. The user must specify the database (and specific tables or arrays) on which the analyses must be run. The former is achieved by writing a Galaxy tool for the relational and array databases that exposes minimal language functionality. Currently, our tool lets users find subsets of data and compute aggregates over them. Probabilistic analysis is enabled through BayesDB,¹¹ but the functionality is limited.

We've also demonstrated how downstream analyses, which are traditionally accomplished by running a parallel MPI job, can now be done within a database. A subsequent step to finding halos is building merger trees, which determine how dark-matter-dominated halos merge over time, beginning with small clumps coming together to form ever larger objects. We wrote a merger tree program as a PDACS Python tool that computes the merger tree over halo properties in SciDB. The rewrite provided an interesting tradeoff between programmatic effort and performance, compared to the original code. The SciDB program was only 50 lines of code, compared to the 650 lines of equivalent C++ code that allocated arrays and read data, computed the merger tree, and wrote it out, but ran five times slower than its corresponding parallel MPI program. For certain types of exploratory analyses, this is a perfectly acceptable tradeoff.

User reaction and feedback will guide future development. Currently, PDACS instances are individually tied to their local computational and

data storage resources. Over time, our aim is to replace these with a single access point from where users can manage the entire analysis—a process that would include the ability to move data from archival sites, if needed. In the future, we expect that PDACS could also be hosted by commercial cloud services, depending on the evolution of the associated service and cost models.

Acknowledgments

We were supported by the US Department of Energy, Basic Energy Sciences, Office of Science, under contract number DE-AC02-06CH11357. Initial support for PDACS development was provided by the US Department of Energy, High Energy Physics. This research used resources at ALCF, Argonne National Laboratory, NERSC, Lawrence Berkeley National Laboratory, and OLCF, Oak Ridge National Laboratory.

References

1. K. Dolag et al., “Simulation Techniques for Cosmological Simulations,” *Space Science Rev.*, vol. 134, 2008, pp. 229–268.
2. G. Lemson et al., “Halo and Galaxy Formation Histories from the Millennium Simulation: Public Release of VO-Oriented and SQL-Queryable Database for Studying the Evolution of Galaxies in the LambdaCDM Cosmogony,” eprint arXiv:astro-ph/060819, 2006.
3. K. Heitmann et al., “The Coyote Universe Extended: Precision Emulation of the Matter Power Spectrum,” *Astrophysical J.*, vol. 780, 2014, article id.111.
4. J. Goecks et al., “Galaxy: A Comprehensive Approach for Supporting Accessible, Reproducible, and Transparent Computational Research in the Life Sciences,” *Genome Biology*, 2010, article id.11:R86.
5. I. Foster, “Globus Online: Accelerating and Democratizing Science through Cloud-Based Services,” *IEEE Internet Computing*, vol. 15, May/June 2011, pp. 70–73.
6. R. Ananthakrishnan et al., “Globus Nexus: An Identity, Profile, and Group Management Platform for Science Gateways and Other Collaborative Science Applications,” *Proc. IEEE Int'l Conf. Cluster Computing*, 2013, pp. 1–3.
7. S. Cholia, D. Skinner, and J. Boverhof, “NEWT: A RESTful Service for Building High Performance Computing Web Applications,” *Proc. 2010 Gateway Computing Environments Workshop*, 2010, pp. 1–11.
8. A. Szalay et al., “The SDSS Skyserver: Public Access to the Sloan Digital Sky Server Data,” *Proc. ACM SIGMOD Int'l Conf. Management of Data*, 2002, pp. 570–581.
9. M. Nieto-Santisteban et al., “When Database Systems Meet the Grid,” *Proc. Conf. Innovative Database Management*, 2005, <http://www-db.cs.wisc.edu/cidr/papers/P13.pdf>
10. M. Stonebraker et al., “SciDB: A Database Management System for Applications with Complex Analytics,” *Computing in Science & Eng.*, vol. 15, 2013, pp. 54–63.
11. J. Baxter, “BayesDB: Querying the Probable Implications of Tabular Data,” PhD dissertation, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 2014.

Ravi Madduri is a project manager in the mathematics and computer science division at Argonne National Laboratory. He has an MS in computer science from the Illinois Institute of Technology. Madduri is a Fellow of the Computation Institute and an Innovation Fellow of the Polsky Center of Entrepreneurship. Contact him at madduri@anl.gov.

Alex Rodriguez is a bioinformatician in the Computation Institute at the University of Chicago and Argonne National Laboratory. He has a PhD in bioinformatics from the University of Illinois. Contact him at arodri7@uchicago.edu.

Thomas Uram is research staff at the Argonne Leadership Computing Facility and the Computation Institute at the University of Chicago. His research interests include large-scale parallel simulation and analysis, next-generation computing architectures, and workflows in support of computational science, recently targeting high-energy physics and computed tomography. Contact him at turam@anl.gov.

Katrin Heitmann is a member of the scientific staff in the high-energy physics and mathematics and computational science divisions at Argonne National Laboratory. Her research interests include physical cosmology, advanced statistical methods, and large-scale computing. Heitmann has a PhD in physics from the University of Dortmund. She's a member of the American Physical Society and a Senior Fellow of the Computation Institute and the Kavli Institute for Cosmological Physics at the University of Chicago. Contact her at heitmann@anl.gov.

Tanu Malik is a research scientist in the Computation Institute and a lecturer in the Department of Computer Science at the University of Chicago. She also holds a staff scientist appointment at Argonne National Laboratory. Her research interests are in the management of large datasets in scientific applications. Malik has a PhD in computer science from Johns Hopkins University. She's a member of ACM and the IEEE Computer Society and a Fellow of the Computation Institute. Contact her at tanum@ci.uchicago.edu.

Saba Sehrish is a computer science researcher at Fermi National Accelerator Laboratory. Her research interests include high-performance parallel I/O libraries, large-scale I/O subsystems, data-intensive scalable computing, and frameworks and I/O for high-energy physics. Sehrish has a PhD in computer engineering from the University of Central Florida. Contact her at ssehrish@fnal.gov.

Ryan Chard is a PhD student in the School of Engineering and Computer Science at Victoria University of Wellington. His research interests include distributed systems, high-performance computing, cloud computing, and reputation. He's a student member of IEEE. Contact him at ryan@ecs.vuw.ac.nz.

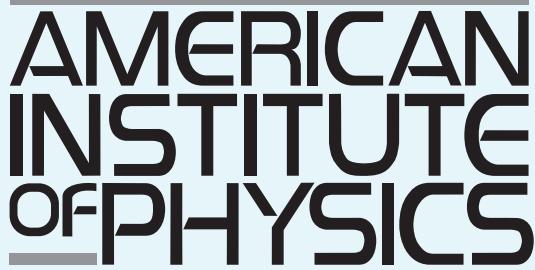
Shreyas Cholia leads the science gateway, Web, and grid efforts at NERSC, with the goal of making high-performance and distributed computing more transparent and accessible. He's particularly interested in how Web APIs and tools can facilitate data-centric computing. Cholia has a BS in computer science and cognitive sciences from Rice University. Contact him at scholia@lbl.gov.

Marc Paterno is a computer science researcher and leader of the Tools and Advanced Computing Group in the Scientific Computing Division at Fermi National Accelerator Laboratory. His research interests include the design and development of frameworks and libraries for

large-scale collection, simulation, and analysis of data in high-energy physics. Paterno has a PhD in physics from the State University of New York at Stony Brook. Contact him at paterno@fnal.gov.

Jim Kowalkowski is a computer science researcher and assistant division head at Fermi National Accelerator Laboratory. His research interests include the evolution and design of large-scale software frameworks and libraries used for high-energy physics experimental data simulations and analysis. Kowalkowski is a member of ACM and the IEEE Computer Society. Contact him at jbk@fnal.gov.

Salman Habib is a senior scientist in the high-energy physics and mathematics and computational science divisions at Argonne National Laboratory. His research interests include quantum and classical dynamical systems and field theory, including the use of large-scale computing resources to solve problems in these fields, and the application of advanced statistical methods and supercomputing to physical cosmology. Habib has a PhD in physics from the University of Maryland. He's a member of the American Physical Society, ACM, and the IEEE Computer Society, a Senior Fellow of the Computation Institute, and a Senior Member of the Kavli Institute for Cosmological Physics at the University of Chicago. Contact him at habib@anl.gov.



The American Institute of Physics is an organization of scientific societies in the physical sciences, representing scientists, engineers, and educators. AIP offers authoritative information, services, and expertise in physics education and student programs, science communication, government relations, career services for science and engineering professionals, statistical research in physics employment and education, industrial outreach, and the history of physics and allied fields. AIP publishes PHYSICS TODAY, the most influential and closely followed magazine of the physics community, and is also home to the Society of Physics Students and the Niels Bohr Library and Archives. AIP

owns AIP Publishing LLC, a scholarly publisher in the physical and related sciences.

Governing Board: *Louis J. Lanzerotti (chair), Samuel H. Aronson, Malcolm R. Beasley, G. Fritz Benedict, J. Daniel Bourland, Charles Carter, Beth Cunningham, Bruce H. Curran, Robert Doering, Judy Dubno, Michael D. Duncan, H. Frederick Dylla (ex officio), David Ernst, Janet Fender, Judith Flippin-Anderson (ex officio), Steve Iona, Marcia Isakson, Paul L. Kelley, Angela R. Keyser, Kate Kirby, Rudolf Ludeke, Jonathan Malay, Kevin B. Marvel, Mary Mogge, Faith Morrison, Homer Neal, Elizabeth A. Rogan, Dian Seidel, Keith Seitter, Joseph Serene, Neal Shinn, Scott Sommerfeldt, Gene Sprouse, Gay Stewart, Hervey (Peter) Stockman, Michael Turner, H. Joe Witte.*
Board members listed in italics are members of the Executive Committee.

Management Committee: H. Frederick Dylla, Executive Director and CEO; Gigi Swartz, Treasurer and CFO; Theresa C. Braun, Vice President, Human Resources; Catherine O'Riordan, Vice President, Physics Resources; Judith Flippin-Anderson, Secretary.