

Workflows using Pegasus: Enabling Dark Energy Survey Pipelines

Karan Vahi,¹ Michael H Wang,² Chihway Chang,³ Scott Dodelson,⁴ Mats Rynge,¹ and Ewa Deelman.¹

¹*USC Information Sciences Institute, Marina Del Rey, California, USA;*
vahi@isi.edu, rynge@isi.edu, deelman@isi.edu

²*Fermi National Laboratory, Batavia, Illinois, USA; mwang@fnal.gov*

³*Kavli Institute of Cosmological Physics - University of Chicago, Chicago, Illinois, USA; chihway@kicp.uchicago.edu*

⁴*Carnegie Mellon University, USA; sdodelso@andrew.cmu.edu*

Abstract. Workflows are a key technology for enabling complex scientific applications. They capture the inter-dependencies between processing steps in data analysis and simulation pipelines, as well as the mechanisms to execute those steps reliably and efficiently in a distributed computing environment. They also enable scientists to capture complex processes to promote method sharing and reuse and provide provenance information necessary for the verification of scientific results and scientific reproducibility. We describe a weak-lensing pipeline that is modelled as a Pegasus workflow with pipeline codes available as a Singularity container. This has enabled us to make this analysis widely available and easily replicable to the astronomy community. Using Pegasus, we have executed various steps of pipelines on different compute sites with varying infrastructures, with Pegasus seamlessly managing the data across the various compute clusters in a transparent manner.

1. Introduction

One of the most exciting and challenging areas of modern cosmology is weak gravitational lensing: the phenomenon of small distortions in the shapes of background galaxies as the light they emit traverses the lumpy universe. By measuring the shapes of galaxies in a given region, cosmologists can infer the total mass along the line of sight. Carried out over large parts of the sky, this inference can shed light on the most profound mysteries in the universe, such as the nature of dark matter and dark energy. The analyses are so complex that the process of making the measurements on petabytes of imaging data, processing the measurement outputs, applying algorithms that measure shapes of billions of galaxies, and then using that information to learn about the universe takes years of effort by large collaborations to carry out. In order to make this analysis widely available and easily replicable to the astronomy community, we decided to model this pipeline as a Pegasus workflow.

Pegasus WMS (Deelman et al. 2015) is a workflow management system that can manage large-scale pipelines across desktops, campus clusters, grids and clouds. In Pegasus WMS, pipelines are represented in an abstract form that is independent of the

resources available to run it and the location of data and executables. It compiles these abstract workflows to executable workflows represented as HTCondor DAG's that can be deployed onto distributed and high-performance computing resources such as DOE LCFs like NERSC, XSEDE, local clusters, and clouds. The executable form is an advanced DAG which is executed by HTCondor DAGMan. Pegasus is used in a number of scientific domains doing production grade science. In 2016 the LIGO gravitational wave experiment used Pegasus for its PyCBC pipeline (Usman et al. 2016) to analyze instrumental data and confirm the first ever detection of a gravitational wave. The Southern California Earthquake Center (SCEC) based at USC, uses a Pegasus managed workflow infrastructure called Cybershake to generate hazard maps for the Southern California region. In March 2017, SCEC conducted a CyberShake study (Callaghan et al. 2017) on DOE systems ORNL Titan and NCSA BlueWaters to generate the latest maps for the Southern California region. Overall, the study required 450,000 node-hours of computation across the two systems. Pegasus is also being used in astronomy, bioinformatics, civil engineering, climate modeling, earthquake science, molecular dynamics and other complex analyses.

2. Benefits of Pegasus Approach for Compute Pipelines

HTCondor DAGMan (Thain et al. 2005) is a common foundation for many astronomy pipelines. When using HTCondor DAGMan directly, astronomy projects commonly find themselves developing pipelines for a particular execution environment and data storage solution, and therefore have to spend valuable development time to continuously adjust the pipeline for new infrastructures or changes to the existing infrastructure. Pegasus WMS solves this problem by providing an abstraction layer on top of HTCondor DAGMan.

Using Pegasus instead of writing out HTCondor DAGs directly, enables us to execute the pipeline on a distributed grid infrastructure, where Pegasus takes care of the data management of the pipeline. During the mapping step, Pegasus WMS tries to look up the LFNs to obtain a list of physical file names (PFN) which are URLs to locations of where the file can be found. For input files, the system determines the appropriate replica to use, and which data transfer mechanism to use. Data transfer tasks are added to the pipeline accordingly. If during the lookups, Pegasus WMS finds that a subset of the pipeline outputs already exists, the pipeline will be automatically pruned to not recompute the existing outputs. This data reuse feature is commonly used for projects with overlapping datasets and pipelines. Required data transfers are automatically added to the pipeline and optimized for performance. Pegasus WMS imports compute environment descriptions provided by the scientist, and URLs for the data to schedule data transfers, including credential management. During the mapping step, it is determined when intermediate data files are no longer required. Clean up tasks are added, with the overall result being a minimized data footprint during execution. Data registration is the feature that adds information about generated outputs to an information catalog for future data discovery and potential data reuse.

In order to keep the science code dependencies portable and easily deployable, Pegasus allows scientists to package their science code and dependencies into a Docker or a Singularity container. The use of application containers ensures the scientific code is executed in a homogenous environment tailored for application, even when executing on nodes with widely varying architecture, operation systems and system libraries. The

container images themselves then managed and deployed on the fly on the nodes where the jobs execute automatically by Pegasus.

3. Weak Lensing Pipeline

The pipeline described here is an example of a typical gravitational weak lensing analysis. It uses publicly available Science Verification catalogs of the Dark Energy Survey (DES-SV).

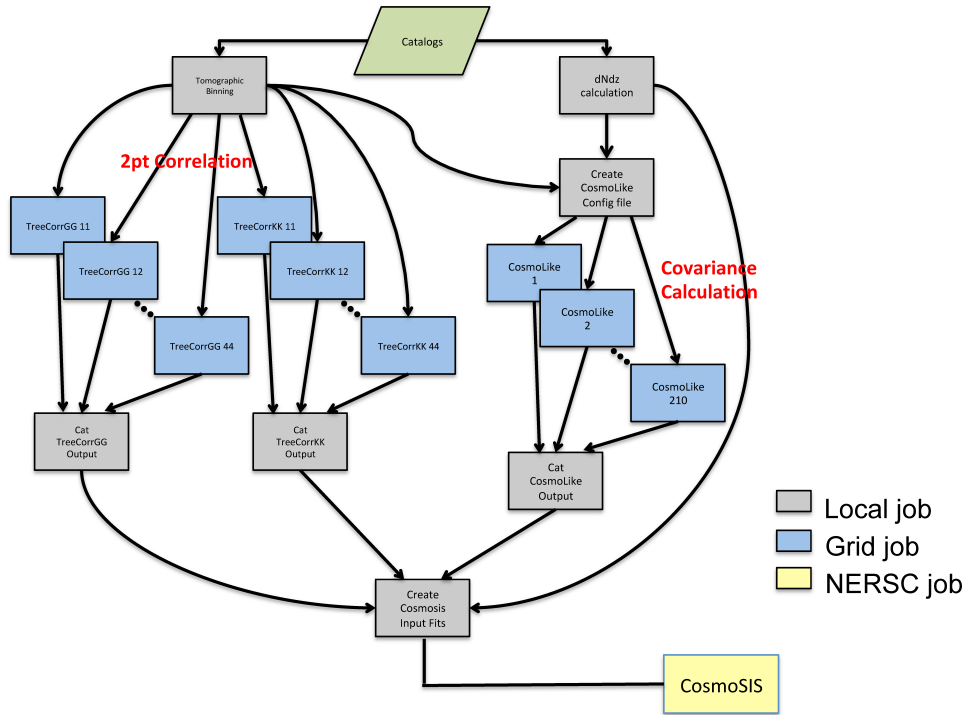


Figure 1. Weak Lensing Pipeline deployed at FNAL

The very first two steps of the pipeline are `doTomoBinning` and `calDndz`, which directly read from the DES-SV input catalogs. `doTomoBinning` selects and sorts the objects in the input shape catalog into tomographic bins, writing out smaller fits files for each bin. `calDndz` sums up the PDFs for each galaxy in the input catalog to calculate the full redshift distribution. After `doTomoBinning` completes, $N_c = nbtomo(nbtomo + 1)/2TreeCorr$ jobs are launched in parallel to calculate the two-point shear correlation functions using the fits files produced by `doTomoBinning` as input.

After both `doTomoBinning` and `calDndz` are done, `CosmoLike` is launched to calculate the analytic covariance associated with the data vector. It uses the redshift distributions from `calDndz` and the effective number density and ellipticity noise from `doTomoBinning` as input. A total of $N_c(2N_c + 1)$ `CosmoLike` jobs are launched in parallel to calculate all the submatrices of the full covariance matrix.

After `TreeCorr` and `CosmoLike` complete, their results are each concatenated into two separate files which, together with the redshift distributions from `calDndz`, serve

as input for `mkCosmosisFits`. This last step produces an output fits file that has all the relevant information arranged in a format supported by the CosmoSIS framework used to extract the cosmological parameters.

We run `mkCosmosisFits` three times since we want to produce three different versions of the FITS files that are used as input for the final stage where we use the CosmoSIS framework to extract the cosmological parameters. The only difference between each version of the FITS file is in the covariance matrix:

- using the original survey-provided covariance (same one used in the Survey’s published results). This is basically for reference.
- using the analytical Gaussian-only covariance from `CosmoLike`
- using the analytical covariance from `CosmoLike` that also include non-Gaussian contributions.

We have used the pipeline described above to carry out a unified analysis of four recent cosmological datasets (DLS, CFHTLenS, DES-SV, KiDS-450). The modular and flexible framework of the pipeline structure allows us to perform the analysis in a systematic way that uses the computational resources effectively and is easy to debug. Pegasus enabled us to execute various steps of pipelines illustrated in Figure 1 on different compute sites (HTCondor pool Grid at FNAL, local workflow server and NERSC), with Pegasus seamlessly managing the data across the various compute clusters in a transparent manner. As part of the unified analysis, about 3,041 HTCondor jobs were executed using about a year’s worth of computing time. A demonstrative version of this pipeline is available on Github (Wang & Vahi 2018) for users to download and perform similar analysis. For the focus demonstration, we used a small HTCondor pool at USC/ISI, configured with 128 slots. The compute nodes had Singularity pre-installed.

Acknowledgments. Pegasus is funded by the National Science Foundation(NSF) under the OAC SI2-SSI #grant 1664162. Previously, NSF has funded Pegasus under OCI SDCI program grant #0722019 and OCI SI2-SSI program grant #1148515.

References

- Callaghan, S., Juve, G., Vahi, K., Maechling, P. J., Jordan, T. H., & Deelman, E. 2017, in 12th Workshop on Workflows in Support of Large-Scale Science (WORKS’17). Funding Acknowledgments: NSF 1664162 and 1443047
- Deelman, E., Vahi, K., Juve, G., Rynge, M., Callaghan, S., Maechling, P. J., Mayani, R., Chen, W., Ferreira da Silva, R., Livny, M., & Wenger, K. 2015, Future Generation Computer Systems, 46, 17. Funding Acknowledgements: NSF ACI SDCI 0722019, NSF ACI SI2-SSI 1148515 and NSF OCI-1053575, URL <http://pegasus.isi.edu/publications/2014/2014-fgcs-deelman.pdf>
- Thain, D., Tannenbaum, T., & Livny, M. 2005, Concurr. Comput. : Pract. Exper., 17, 323. URL <http://dx.doi.org/10.1002/cpe.v17:2/4>
- Usman, S. A., Nitz, A. H., Harry, I. W., Biwer, C. M., Brown, D. A., Cabero, M., Capano, C. D., Canton, T. D., Dent, T., Fairhurst, S., Kehl, M. S., Keppel, D., Krishnan, B., Lenon, A., Lundgren, A., Nielsen, A. B., Pekowsky, L. P., Pfeiffer, H. P., Saulson, P. R., West, M., & Willis, J. L. 2016, Classical and Quantum Gravity, 33, 215004. URL <http://stacks.iop.org/0264-9381/33/i=21/a=215004>
- Wang, M. H., & Vahi, K. 2018, Executing Weak Lensing Pipelines using Pegasus, <https://github.com/pegasus-isi/pegasus-wlpipe>