Volume Title
*ASP Conference Series, Vol. **Volume Number***
Author

Producing an Infrared Multiwavelength Galactic Plane Atlas using Montage, Pegasus and Amazon Web Services

Mats Rynge[1], Gideon Juve[1], Jamie Kinney[2], John Good[3], Bruce Berriman[3], Ann Merrihew[2], and Ewa Deelman[1]

[1]*Information Sciences Institute, University of Southern California, 4676 Admiralty Way, Suite 1001, Marina del Rey, CA 90292, USA*

[2]*Amazon Web Services, 1918 8th Ave., Seattle, WA 98101, USA*

[3]*Infrared Processing and Analysis Center, California Institute of Technology, 770 South Wilson Avenue, Pasadena, CA 91125, USA*

Abstract. In this paper, we describe how to leverage cloud resources to generate large-scale mosaics of the galactic plane in multiple wavelengths. Our goal is to generate a 16-wavelength infrared Atlas of the Galactic Plane at a common spatial sampling of 1 arcsec, processed so that they appear to have been measured with a single instrument. This will be achieved by using the Montage image mosaic engine process observations from the 2MASS, GLIMPSE, MIPSGAL, MSX and WISE datasets, over a wavelength range of 1 μm to 24 μm, and by using the Pegasus Workflow Management System for managing the workload. When complete, the Atlas will be made available to the community as a data product.

We are generating images that cover ±180° in Galactic longitude and ±20° in Galactic latitude, to the extent permitted by the spatial coverage of each dataset. Each image will be 5°x5° in size (including an overlap of 1° with neighboring tiles), resulting in an atlas of 1,001 images. The final size will be about 50 TBs.

This paper will mainly focus on the computational challenges, solutions and lessons learned in producing the Atlas. To manage the computation we are using the Pegasus Workflow Management System, a mature, highly fault-tolerant system now in release 4.2.2 that has found wide applicability across many science disciplines. A scientific workflow describes the dependencies between the tasks and in most cases the workflow is described as a directed acyclic graph, where the nodes are tasks and the edges denote the task dependencies. A defining property for a scientific workflow is that it manages data flow between tasks. Applied to the galactic plane project, each 5 by 5 mosaic is a Pegasus workflow. Pegasus is used to fetch the source images, execute the image mosaicking steps of Montage, and store the final outputs in a storage system.

As these workflows are very I/O intensive, care has to be taken when choosing what infrastructure to execute the workflow on. In our setup, we choose to use dynamically provisioned compute clusters running on the Amazon Elastic Compute Cloud (EC2). All our instances are using the same base image, which is configured to come up as a master node by default. The master node is a central instance from where the workflow can be managed. Additional worker instances are provisioned and configured to accept work assignments from the master node. The system allows for adding/removing workers in an ad hoc fashion, and could be run in large configurations.

To-date we have performed 245,000 CPU hours of computing and generated 7,029 images and metadata totaling 30 TB. With the current set up our runtime would be 340,000 CPU hours for the whole project. Using spot m2.4xlarge instances, the cost would be approximately $5,950. Using faster AWS instances, such as cc2.8xlarge could potentially decrease the total CPU hours and further reduce the compute costs. The paper will explore these tradeoffs.

1. Introduction

While this paper is mainly focusing on the infrastructure and software stack used, it is important to note that there is a real science goal driving the effort. The final product is a multiwavelength image atlas of the galactic plane, composed of images at 16 different wavelengths from 1 μm to 24 μm, processed so that they appear to have been measured with a single instrument observing all 16 wavelengths.

Survey / Bands (μm)	Coverage of 360°x40° area	Output Size (TB)	Compute time (1,000s core hours)
2MASS (1.2, 1.6, 2.2)	100%	14.4	87
GLIMPSE (3.6, 4.5, 5.8, 8.0)	11%	2.0	60
MIPSGAL (24)	8%	0.4	3
MSX (8.8, 12.1, 14.6, 21.3)	35%	6.8	36
WISE (3.4, 4.6, 12, 22)	100%	19.2	132

Table 1. The survey/bands included in the Galactic Plane dataset, including output data sizes, and compute times on hi1.4xlarge instances.

We used the Montage (Jacob et al. 2009) image mosaic engine to transform all the images in the surveys to a common pixel scale of 1 second or arc, where all the pixels are co-registered on the sky and represented in galactic coordinates and the Cartesian projection. The final dataset covers 360° along the galactic plane and ±20° on either side of it. Each output image is 5° by 5 °in size, and has an overlap of 1° with neighboring tiles. With this setup, and starting with a tile centered at 0 latitude, 0 longitude, each of the 16 target dataset/band combinations could potentially have 1,001 images. However, some of the input datasets did not provide full coverage of the targeted region, and for those cases only the area covered by the input dataset was processed. Table 1 provide a list of datasets, coverage, data sizes, and compute time required for the processing. The total output dataset size is 45 TB. When complete, the data will be released to the community via an API.

2. Galactic Plane Workflow

To manage the tasks required to generate all the images, 16 hierarchical Pegasus workflows were created, one for each survey/band. The Pegasus Workflow Management System (WMS) (Deelman et al. 2005) is used by scientists to execute large-scale computational workflows on a variety of cyberinfrastructure, ranging from local desktops to campus clusters, grids, and commercial and academic clouds. Pegasus WMS enables scientists to compose abstract workflows without worrying about the details of the underlying execution environment or the particulars of the low-level specifications required by the middleware. The run a workflow, the scientist provides an abstract workflow, a list of available software (transformation catalog), a list of input files (replica

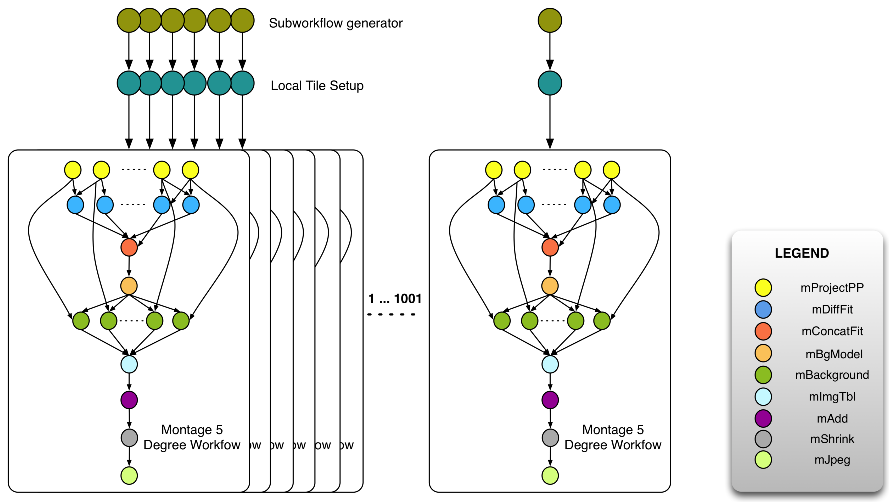

Figure 1. One instance of the Galactic Plane hierarchical workflow. Each survey/band workflow contains 1,001 subworkflows.

catalog), and a description of available execution environments (site catalog) to the Pegasus planner. The output of the planning step is an executable workflow, which includes workflow transformations such as performance optimization and added data management tasks, which can then be executed using HTCondor (Thain et al. 2005) for job management.

In each hierarchical workflow, the top level workflow manages the data find tasks against the IPAC archives, and then creates the 1,001 sub workflows needed to produce the tiles. Each subworkflow uses Montage to repoject the input images to the coordinate space of the requested mosaic. The reprojected images are then background rectified and finally coadded to create the final output mosaic. Figure 1 shows the layout of the hierarchical workflow. Not counting reruns, it took 16,022 workflows to complete the work.

3. Amazon Web Services

Amazon Web Services (AWS) provides a cloud infrastructure with effectively limitless capacity. While the sheer scale of available compute capacity initially attracts some researchers to AWS, this project benefited greatly from the flexibility provided by Amazon S3 object storage. S3 serves both as a high performance temporary space for workflow execution, as well as the long term reliable solution for durably storing and sharing the final dataset.

In our setup, compute clusters can be created easily using Amazon EC2 with both short term and long term storage provided by Amazon S3. It is common for parallel applications running on EC2 to take advantage of auto-scaling, i.e. have the compute cluster grow and shrink based on compute demand. However, the Galactic Plane workflow is data constrained by IPAC's available network bandwidth, and therefore we choose a more static setup in which compute nodes can be added and removed via the

AWS web console or APIs. The majority of the time, we ran on a smaller sized cluster than optimal, just to not overwhelm IPAC data find and data transfer services.

The requirement for our instances is that there needs to be one master instance, and a variable set of worker instances associated with the master instance. This master/worker setup stems from the logical setup of the job scheduling software used, HTCondor, which has a submit/central node (our master instances) and a set of job execution nodes (our worker instances). The master instance is the one a user can log into, and submit and manage workflows. To make maintaining the virtual machine image easy, both the master and the worker instances are using the same base image. The image is configured to come up as a master node by default. If an IP address of a master node is given to instance at boot time, the boot scripts configure the instance as a worker and attaches itself to the master specified. The image is setup to use two ephemeral disks in a striped RAID-0 configuration to increase the disk I/O performance.

The jobs in the workflow require a centralized storage system to retrieve and store intermediate and output files. Pegasus is flexible in this regard, and can work against a set of different storage protocols, including S3 used for the Galactic Plane workflow. S3, being an object store, scales very well with the number of clients, but has some latency associated with each request. In order to better overlap computation and data retrieval/storing, HTCondor was configured to oversubscribe the worker nodes with jobs. By default, HTCondor detects the number of cores on a node, and exposes the same number of job slots. The number of jobs slots configuration was changed on our worker nodes to be the number of detected cores + 2, which resulted in close to a 100% CPU utilization.

Using the hi1.4xlarge instance type, the project took 318,000 core hours to complete. If AWS had not provided the processing, the cost using spot pricing would be approximately $5,950. Comparison tests show that if we had chosen to run the workload on cc2.8xlarge instances with spot pricing, it would reduce the total cost to $2,200 based on the lower cost per core/hour and faster processor speed. These costs do not include data storage.

Acknowledgements

Compute resources for the project were provided by an AWS in Education Research Grant (`http://aws.amazon.com/grants/`). Storage for the Galactic Plane data set is provided by the AWS Public Data Sets program (`http://aws.amazon.com/datasets/`). The National Science Foundation funds Pegasus under the OCI SI2-SSI #1148515 grant. G. B. Berriman and J. C. Good are supported by the NASA Exoplanet Science Institute at the Infrared Processing and Analysis Center, operated by the California Institute of Technology in coordination with the Jet Propulsion Laboratory (JPL).

References

Deelman, E., Singh, G., Su, M.-H., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., Vahi, K., Berriman, G. B., Good, J., et al. 2005, Scientific Programming, 13, 219

Jacob, J. C., Katz, D. S., Berriman, G. B., Good, J. C., Laity, A. C., Deelman, E., Kesselman, C., Singh, G., Su, M., Prince, T. A., & Williams, R. 2009, Int. J. Comput. Sci. Eng., 4, 73. URL `http://dx.doi.org/10.1504/IJCSE.2009.026999`

Thain, D., Tannenbaum, T., & Livny, M. 2005, Concurrency - Practice and Experience, 17, 323