Montage: The Architecture and Scientific Applications of a National Virtual Observatory Service for Computing Astronomical Image Mosaics

G. B. Berriman, A. C. Laity, J. C. Good
Infrared Processing and Analysis Center, California Institute of Technology
Pasadena, CA 91125

J. C. Jacob and D. S. Katz
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA 91109

E. Deelman, G. Singh, and M.-H. Su
USC Information Sciences Institute,
Marina del Rey, CA 90292

T. A. Prince
Division of Physics, Mathematics and Astronomy, California Institute of Technology,
Pasadena, CA 91125

***Abstract* - Montage is a portable toolkit for constructing custom, science-grade mosaics by composing multiple astronomical images. The mosaics constructed by Montage preserve the astrometry (position) and photometry (intensity) of the sources in the input images. The mosaic to be constructed is specified by the user in terms of a set of parameters, including dataset and wavelength to be used, location and size on the sky, coordinate system and projection, and spatial sampling rate. Many astronomical datasets are massive, and are stored in distributed archives that are, in most cases, remote with respect to the available computational resources. The paper describes scientific applications of Montage by NASA projects and researchers, who run the software on both single- and multi-processor computers, including clusters and grids. Standard grid tools are used to run Montage in the case where the data or computers used to construct a mosaic are located remotely on the Internet. This paper describes the architecture, algorithms, and performance of Montage as both a software toolkit and as a grid portal.**

I. INTRODUCTION

Wide-area imaging surveys have assumed fundamental importance in astronomy. They are being used to address such fundamental issues as the structure and organization of galaxies in space and the star formation history of our galaxy. One of the most powerful probes of the structure and evolution of astrophysical sources is the variation of their properties with wavelength, but this power has yet to be fully realized in the analysis of astrophysical images because survey results are published in widely varying coordinates, map projections, sizes and spatial resolutions. Moreover, the spatial extent of many astrophysical sources is much greater than that of individual images. Astronomy therefore needs a general image mosaic engine that will process input images to deliver image mosaics of arbitrary size in any common coordinate system, in any map projection and at any spatial sampling rate. The Montage project ([1], [2], [3]) has developed this capability as a scalable, portable toolkit that can be used on desktops for science analysis, integrated into project and mission pipelines, or run on computing grids to support large-scale product generation, mission planning and quality assurance. This paper reviews the design of Montage, its scalability, its deployment on the TeraGrid (MontageTG), its performance and its scientific applications.

II. DESIGN

Montage's goal is to provide astronomers with software for the computation of custom science-grade image mosaics in the Flexible Image Transport System (FITS) format, the standard format in astronomy. Custom refers to user specification of mosaic parameters, including World Coordinate System (WCS) projection, coordinate system, mosaic size, image rotation, and spatial sampling rate. Science-grade mosaics preserve the calibration and astrometric (spatial) fidelity of the input images.

There are three steps to building a mosaic with Montage:

- Reprojection of input images to a common projection, coordinate system, and spatial scale,
- Modeling of background radiation in images to rectify them to a common flux scale and background level, thereby minimizing the inter-image differences, and
- Coaddition of reprojected, background-rectified images into a final mosaic.

Montage accomplishes these tasks in independent, portable, ANSI C modules. This approach controls testing and maintenance costs, and provides flexibility to users. They can, for example, use Montage simply to reproject sets of images and co-register them on the sky, implement a custom background removal algorithm, or define another processing flow through custom scripts. Table 1 lists the core computational Montage modules.

Table 1: The core design components of Montage

Component	Description
MOSAIC ENGINE COMPONENTS	
mImgtbl	Extract geometry information from a set of FITS headers and create a metadata table from it.
mProject	Reproject a FITS image.
mProjExec	A simple executive that runs *mProject* for each image in an image metadata table.
mAdd	Coadd the reprojected images to produce an output mosaic.
BACKGROUND RECTIFICATION COMPONENTS	
mOverlaps	Analyze an image metadata table to determine which images overlap on the sky.
mDiff	Perform a simple image difference between a pair of overlapping images. This is meant for use on reprojected images where the pixels already line up exactly.
mDiffExec	Run *mDiff* on all the overlap pairs identified by *mOverlaps*.
mFitplane	Fit a plane (excluding outlier pixels) to an image. Meant for use on the difference images generated by *mDiff*.
mFitExec	Run *mFitplane* on all the *mOverlaps* pairs. Creates a table of image-to-image difference parameters.
mBgModel	Modeling/fitting program which uses the image-to-image difference parameter table to interactively determine a set of corrections to apply to each image to achieve a "best" global fit.
mBackground	Remove a background from a single image (a planar correction has proven to be adequate for the images we have dealt with).
mBgExec	Run *mBackground* on all the images in the metadata table

Three usage scenarios for Montage are as follows: the modules listed in Table 1 may be run as stand-alone programs; the executive programs listed in the table (i.e., *mProjExec*, *mDiffExec*, *mFitExec*, and *mBgExec*) may be used to process multiple input images either sequentially or in parallel. Two instances of parallel technology have been investigated by Montage: one is MPI; the second is Planning and Execution for Grids (Pegasus) [4]. In either instance, the design of Montage permits the same set of core compute modules to be used regardless of the computational environment being used. Among the benefits of Pegasus is that it allows various scheduling techniques to be used to optimize the concrete workflow for a particular platform, and makes best use of available resources through dynamic workflow mapping. Consequently it has been implemented as part of an architecture that supports on-request mosaic production on the TeraGrid.

III. MONTAGE ON THE TERAGRID

This section describes an infrastructure to support on-request compute services on the TeraGrid. Montage is, in this context, an exemplar compute engine with a general architecture, and lightweight, reusable components. Users need only submit a request through a client such as a web browser and wait for notification that the mosaic has been computed. The underlying architecture is responsible for submitting the job, monitoring it, reporting progress to the users, and of course, notifying the user that the mosaic is ready for pickup.

Figure 1 illustrates the architectural components, itemized below:

- Compute application: Montage. The Montage image mosaic toolkit has been installed on the TeraGrid nodes and runs under the auspices of Condor to produce the mosaic product.

- Request Management Environment (ROME). Developed at IPAC with NVO funding, ROME is a set of Enterprise Java Beans (EJBs) which accept processing requests from users (via servelets), manage processing queues which ensure that resources distribute processing among users and properly load compute resources, and handle monitoring and user notification for jobs that may take hours or days to complete.

- Browser (or other client). Many types of interface can be supported by ROME, but the initial focus is a simple form and/or URL GET.

- Pegasus and the Globus toolkit. The Montage application is easily parallelized and has been implemented via a Condor DAG (Directed Acyclic Graph), a standard approach for TeraGrid processing. We use the Pegasus software developed by the Information Sciences Institute (ISI) to process an easily readable abstract DAG (which we generate with custom code) into a Condor-specific DAG targeting specific TeraGrid resources. Pegasus in turn utilizes the Globus Replica Location Service (RLS) tool for finding the true physical location of file resources given a more abstract "name".

- Application-specific middleware. The specific computations done in each processing step, and the interdependence of the steps are the only parts of this architecture that are custom to our mosaicking application. It is implemented in two parts: a Montage module, mDAG, which generates an abstract DAG given a list of input images and a CGI-like wrapper program which processes the input parameters, sets up storage space for the final result, queries remote resources (SIA services) to obtain a list of images, runs the DAG builder and Pegasus, and finally submits the Condor-specific DAG for TeraGrid processing.

- TeraGrid tools. Once submitted to Condor on the TeraGrid, the individual processing steps in the DAG are scheduled, data are moved as needed, and the Montage modules are called. This environment marshals compute resources and intermediate file storage space, handles errors by rescheduling subtasks, and reports results back through the submitting machine to ROME and thence to the user.

- SRB and GPSF-WAN. In order to avoid overloading I/O resources that are unable to keep up with TeraGrid processing, we are using a copy of the 2MASS data stored at SDSC in a distributed file system (/gpfs-wan) as input and putting our results in a URL-accessible location within SDSC's Storage Resource Broker (SRB) system. As SRB will also be one of the first full-scale VOSpace implementations, this will become the first large-scale use cases for that technology.

This infrastructure is capable of supporting asynchronous web services, X-509 certificates, and secure transactions (though none of that is currently in place for the initial services).

This set of tools is by design general and easily augmented. The backend could just as easily be a local Condor pool or even a single processor machine (and versions of the services are in fact implemented in this way). ROME is not tied to this specific processing scenario, since it just deals with requests and notification. And a DAG, once generated, could be submitted and monitored manually.

We plan to augment the above in two ways: with additional datasets (SDSS, DPOSS, etc.) as they become available through high-end computing pathways (/gpfs-wan, GridFTP, or SRB) and by adding more specific processing scenarios. The first two of these will be list-driven cutouts for large numbers of sources and mosaicking of user-supplied data. Both are simple variants on the MontageTG / mDAG construct above and can be implemented very quickly.

IV. PERFORMANCE

The TeraGrid portal exploits Pegasus' capability to adapt the workflow to the resources available. But how does its performance compare with MPI implementation? The performance of Montage under both MPI and Pegasus has been measured by a benchmark problem that generates a mosaic of 2MASS data from a 6 x 6 degree region at M16. This requires 1,254 input 2MASS images, each about 0.5 megapixel, for a total of about 657 megapixels (about 5 GB with 64 bits/pixel double precision floating point data). The output is a 3.7 GB FITS (Flexible Image Transport System) file with a 21,600 x 21,600 pixel data segment, and 64 bits/pixel double precision floating point data. The output data is a little smaller than the input data size because there is some overlap between neighboring input images. For the timing results reported in this section, the input data had been pre-staged to a local disk on the compute cluster.

Results in this paper have been measured on the "Phase 2" TeraGrid cluster at the National Center for Supercomputing Applications (NCSA), unless otherwise mentioned. This cluster had (at the time of the experiment) 887 nodes, each with dual Itanium 2 processors with at least 4 GB of memory. 256 of the nodes have 1.3 GHz processors, and the other 631 nodes have 1.5 GHz processors. The timing tests reported here used the 1.5 GHz processors. The network between nodes is Myrinet and the operating system is SuSE Linux. Disk I/O is to a 24 TB General Parallel File System (GPFS). Jobs were scheduled on the system using Portable Batch System (PBS) and the queue wait time was not included in the execution times since that is heavily dependent on machine load from other users.

When using remote grid resources for the execution of the concrete workflow, there is a non-negligible overhead involved in acquiring resources and scheduling the computation over them. To reduce this overhead, Pegasus can aggregate the nodes in the concrete workflow into clusters so that the remote resources can be utilized more efficiently. The benefit of clustering is that the scheduling overhead (from Condor-G, DAGMan and remote schedulers) is incurred only once for each cluster. In the following results, we clustered the nodes in the workflow within a workflow level (or workflow depth). In the case of Montage, the *mProject* jobs are within a single level, *mDiff* jobs are in another level, and so on. Clustering can be done dynamically based on the estimated run time of the jobs in the workflow and the processor availability.

Figure 2 shows the end-to-end time taken to create (*mDAG* and Pegasus) and execute (runtime) the concrete workflow to construct a 6 x 6 degree mosaic. As previously mentioned, Condor Glidein was used to acquire the resources. Once the resources are acquired, they were available for executing the workflow and there was no queuing delay at the remote resource. The workflow was executed using DAGMan running on a host at USC Information Sciences Institute. The time taken to transfer the input data and the output mosaic is not included in this figure. These measurements were made using Montage version 3.0β5. In this version, *mDiff* and *mFitplane* are also available as a single module called *mDiffFit*, which has been used in the timing results shown.

The figure shows the time in minutes for DAGMan to execute the workflow for different numbers of processors. The nodes in the workflow were clustered so that the number of clusters at each level of the workflow was equal to the number of processors. As the number of processors was increased (and thus the number of clusters increases), the Condor overhead becomes the dominant factor. DAGMan takes approximately 1 second to submit each cluster into the Condor queue. Condor's scheduling overhead adds additional delay. As a result we do not always see a corresponding decrease in the workflow execution time as we increase the number of processors. Also, as with the MPI results, the other codes running on the test machine appear to impact these timings. The 64-processor case seems to have worse performance than the 32-processor case, but it is likely that were it rerun on a dedicated machine, it would have better performance. Finally, there are sequential sections in the workflow that limit the overall parallel efficiency.

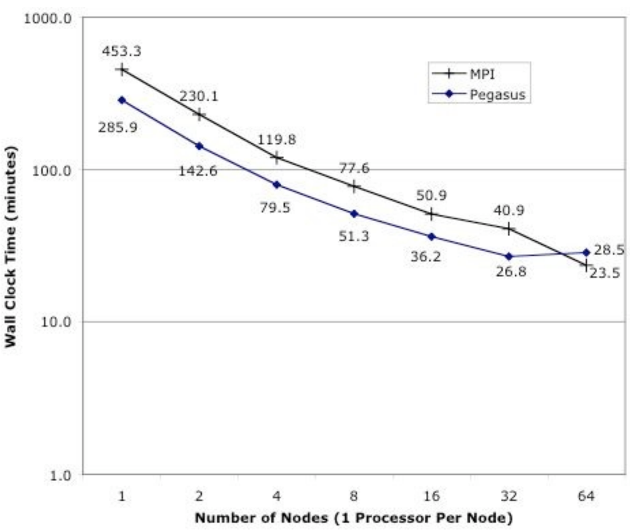

Figure 2: Times for building and executing the concrete workflow for creating a 6 x 6 degree mosaic.

Figure 3 shows a comparison of the time for the MPI run vs. the time needed to build and run the concrete DAG, for the benchmark problem. Notice that the performance of the Pegasus version seems to be faster than the MPI version except at 64 processors where the results are reversed. It is the authors' belief that, for large jobs, the measured difference between the Pegasus and MPI runs is not significant, and that it is due to the I/O contention caused by other jobs running on the test platform during these experiments.

In summary, the MPI versions of the computation-intensive modules perform well but are somewhat limited in the usefulness. A second alternative, using Pegasus and other grid tools, is more general and allows for execution on a variety of platforms without requiring a change in the underlying code base, and appears to have real-world performance comparable to that of the MPI approach for reasonably large problems.

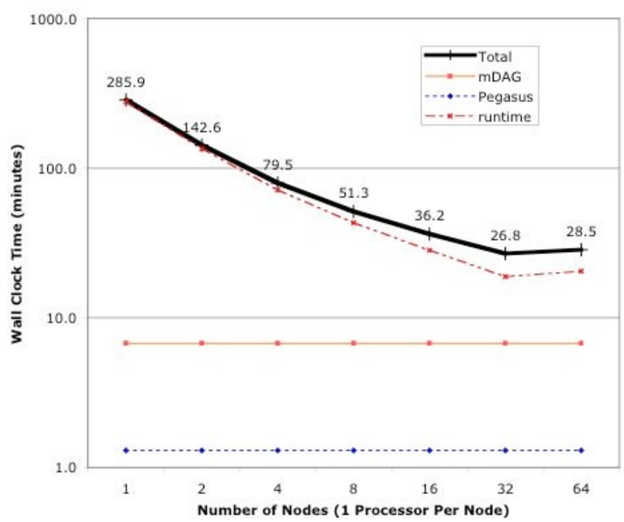

Figure 3: Times for building and executing the concrete workflow for creating a 6 x 6 degree mosaic.

V. SCIENTIFIC APPLICATIONS

This section describes the application of Montage to scientific product generation and product quality assurance, the deployment of on-line data query services, and analysis of astronomical data.

A. Scientific Product Generation and Product Quality Assurance.

The Spitzer Space Telescope supports Legacy programs, whose data products are of exceptional long-term value to astronomy. Two of these projects, the Spitzer Wide-area Infrared Extragalactic Survey (SWIRE) [5] and the Galactic Legacy Infrared Mid-Plane Survey Extraordinaire (GLIMPSE) [6], have incorporated Montage into their data processing pipelines. These pipelines are generating scientific data products that are being made publicly accessible through the NASA/IPAC Infrared Science Archive (IRSA) [7] and the Spitzer Space Telescope archive [8].

The SWIRE has imaged 49 square degrees of the sky (equivalent to the area covered by about 250 full moons), covering six different regions. The SWIRE Spitzer data products have been measured with the Infrared Array Camera (IRAC) (3.6, 4.5, 5.8 and 8.0 µm) and the Multiband Imaging Photometer (MIPS) (24, 70 and 160 µm), supported by optical ancillary data for selected regions. The optical data in most cases is u, g, b, r, i, z bands and originates either from the 2.5m Isaac Newton Telescope (INT) in La Palma, KPNO, or CTIO. These ancillary data come from different telescopes, have different resolutions, orientation, projections and are in a variety of file formats, and so significant effort is put towards converting these data to a uniform base set, which makes data analysis much easier for the SWIRE team and for the public. Montage has been incorporated into the SWIRE processing system to mass-process thousands of optical observations to a common set of image parameters with the Spitzer observations; these parameters are projection, orientation coordinate system, spatial sampling and tiling scheme. The optical and Spitzer data are then fed into multi-band visualization tools to find new sources across the optical, near infrared and infrared wavelength bands. Figure 4 shows one example of a multi-wavelength image, visual and infrared observations of the Tadpole Galaxy.

GLIMPSE is surveying the plane of the galaxy with the Infrared Array Camera (IRAC) aboard Spitzer. It will provide the first global survey of star formation in the Galaxy. The project is changing our view of the Galaxy. It has, for instance, discovered a bar in our Galaxy 27,000 light years long. The principal data products are image mosaics of IRAC measurements of the Galactic Plane over 220 sq deg in four colors. Montage has been integrated as a reprojection engine in the GLIMPSE mosaic pipeline, a cluster of Linux machines.

Montage has been used by SWIRE and GLIMPSE to support quality assurance of the data products described above. SWIRE has used Montage as a fast reprojection and co-addition engine to build sky simulations at a common spatial sampling that model the expected behavior of the sky, including galaxies, stars and cirrus. These simulations have been used to validate the processing pipeline and source extraction. Predictions of the expected source populations and appearance of the sky have been used to plan the observing strategy.

As part of their quality assurance program, GLIMPSE has developed mosaics of their entire survey region at J (1.25µm), H (2.2µm), and K (2.2 µm) and MSX 8µm. They provide quick-look comparisons for quality assurance of the IRAC mosaics.

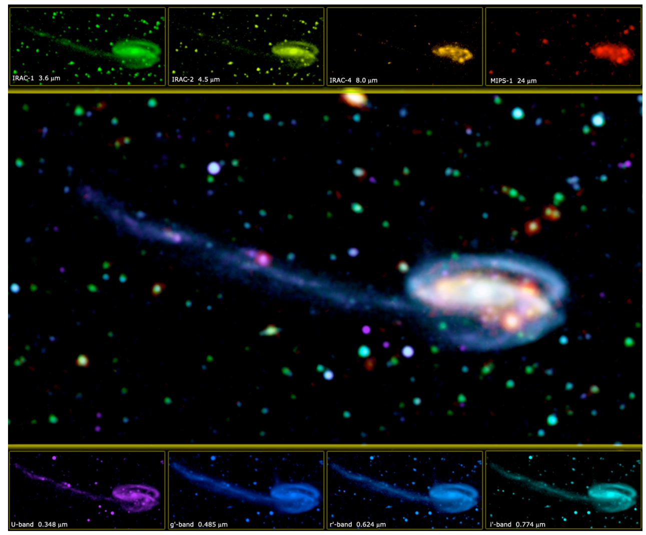

Figure 4: Multi-wavelength Image Mosaic of the Tadpole Galaxy (Courtesy: Dr. Carol Lonsdale)

B. On-Line Data Query Services.

Figure 5 shows an image of the 100 µm map of the sky that aggregates the sky maps produced by the *Diffuse Infrared Background Experiment* (DIRBE), aboard the *Cosmic Background Explorer* (COBE), and the *Infrared Astronomical Satellite* (IRAS) [9]. The map is shown transformed from the Zenithal Equal Area projection as published [5] to the Cartesian projection. This map is a science product that is accessible to astronomers through an on-line service at IRSA. The map is of particular value to astronomers because 100 µm emission can be used to estimate the extinction by interstellar dust along the line through the Galaxy. The on-line service returns cutouts of the map at requested positions, along with estimates of dust emission, galactic emission and extinction along a line of sight. This service will be extended to form an observation planning service required by the Herschel mission, and will specifically support estimate of dust emission extrapolated from 100 µm to the wavelengths at which Herschel will observe.

The Montage reprojection engine has also been integrated into IRSA's on-line Finder Chart service. It provides a visualization tool that cross-compares image data sets from three large-area sky surveys: the Two Micron All Sky Survey (2MASS), the First and Second Generation Digitized Sky Survey, and the Sloan Digital Sky Survey (SDSS). This service enables users to explore a piece of sky taken at different wavelengths and at different times. The Montage reprojection engine is used to place images from these surveys on a common set of image parameters, and the computed images are then made into 3-color images. Because the surveys have been made at different times, the images have proven are a powerful discriminator between image artifacts and astronomical sources, and between fast-moving objects and background stars.

As part its data ingestion and validation process, IRSA is using *mImgTbl*, which extracts geometrical information from a collection of FITS images,. This module is valuable in assessing whether image data sets delivered to the archive contain metadata to fully describe the footprints of the image on the sky. This information is required to support queries that return information on sky coverage of images in a particular part of the sky.

C. Scientific Analysis

Montage is in use by astronomers [10] in studying the large-scale structure of molecular clouds and star forming regions. It is enabling studies of the large-scale distribution of the mass of the clouds via extinction mapping and the positions and evolutionary status of young stellar objects over the extent of the molecular clouds. When placed on a common set of parameters, images at many wavelengths, from the optical to the far infrared, support a global analysis of molecular clouds.

REFERENCES

[1]The Montage website*, http://montage.ipac.caltech.edu*

[2] Berriman, G. B., Curkendall, D., Good, J. C., Jacob, J. C., Katz, D. S., Kong, M., Monkewitz, S., Moore, R., Prince, T. A. and Williams, R. E. (2002) *An architecture for access to a compute intensive image mosaic service in the NVO* in Virtual Observatories, A S. Szalay, ed., Proceedings of SPIE, v. 4846: 91-102.

[3] Berriman, G. B., Deelman, E., Good, J. C., Jacob, J. C., Katz, D. S., Kesselman, C., Laity, A. C., Prince, T. A., Singh, G. and Su, M.-H. (2004) *Montage: A grid enabled engine for delivering custom science-grade mosaics on demand*, in Optimizing Scientific Return for Astronomy through Information Technologies, P. J. Quinn, A. Bridger, eds., Proceedings of SPIE, 5493: 221-232.

[4] Deelman, E., Singh, G., Su, M. – H., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., Vahi, K., Berriman, G. B., Good, J., Laity, A., Jacob, J. C., Katz, D, S. *"Pegasus: a Framework for Mapping Complex Scientific Workflows onto Distributed Systems,"* Journal of Scientific Programming, 13(3), November 2005.

[5] The SWIRE website, www.ipac.caltech.edu/SWIRE/

[6]The GLIMPSE website, http://www.astro.wisc.edu/sirtf/

[7] The NASA/IPAC Infrared Science Archive website, http://irsa.ipac.caltech.edu

[8] The Spitzer Space Telescope Archive, http://ssc.spitzer.caltech.edu/archanaly/

[9] Schlegel, D. J, Finkbeiner, D.P, and Davis, M. (1998), *Ap. J.,* 500, 525

[10] Ciardi, D. R. 2006, In preparation.

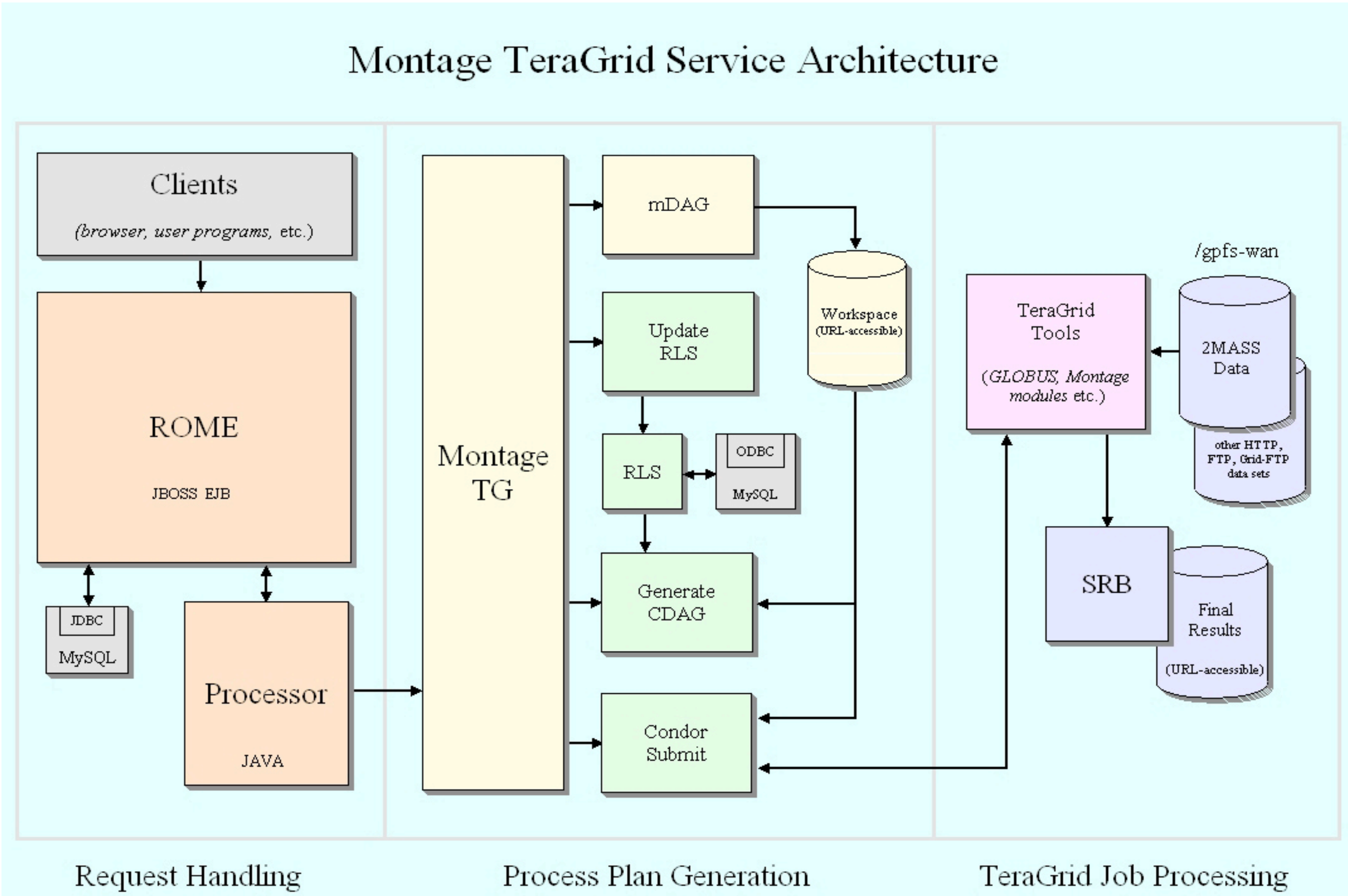

Figure 1: The Montage TeraGrid Service Architecture

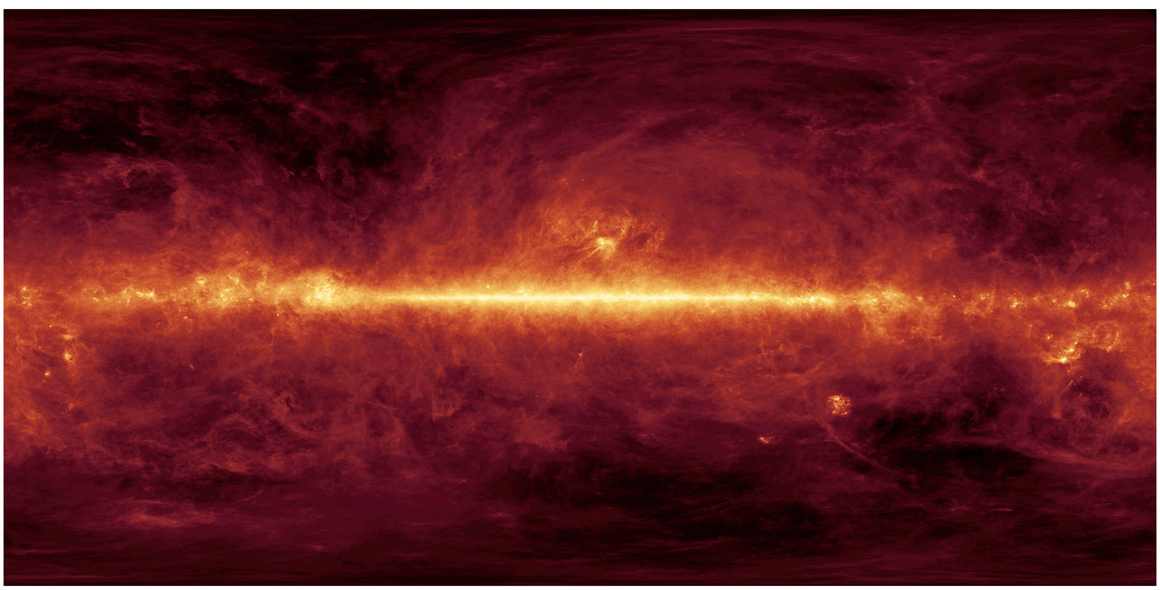

Figure 5: The 100 μm sky represented in Cartesian projection, computed by Montage from composite DIRBE and IRAS sky maps in [5].

Montage is funded by NASA's Earth Sciences Technology Office, Computational Technologies (ESTO-CT), under Cooperative Agreement Number NCC5-626 between NASA and the California Institute of Technology. Pegasus is supported by thee NSF under grants ITR-0086044 (GriPhyn) and ITR AST0122449 (NVO).