An Architecture for Processing Image Mosaics with Montage on the TeraGrid

Prepared by Joseph Jacob
On behalf of the Montage team

January 13, 2004

1 Purpose of this Document

Montage is a portable toolkit that processes images serially or in parallel; it is designed so it can run on any parallel environment, including a Linux cluster or a "true" grid. Montage was designed with three needs in mind: (1) portability; (2) flexibility to the user; (3) adaptability to any processing environment. A computational Grid can take many different forms including a collection of supercomputers, cluster computers, or a pool of workstations connected by a network. This document describes the design of one implementation of a grid portal for Montage. This work is in fulfillment of Milestone I, (see the complete list of milestones at http://montage.ipac.caltech.edu/ms.html). This milestone calls for a prototype architecture that accepts requests for a custom 2MASS image mosaic through a web portal, processes the requests on the TeraGrid (described below), and returns the image mosaic for visualization and analysis.

The TeraGrid web site at http://teragrid.org/: describes the TeraGrid as follows:

> *TeraGrid is a multi-year effort to build and deploy the world's largest, fastest, distributed infrastructure for open scientific research. When completed, the TeraGrid will include 20 teraflops of computing power distributed at five sites, facilities capable of managing and storing nearly 1 petabyte of data, high-resolution visualization environments, and toolkits for grid computing. These components will be tightly integrated and connected through a network that will operate at 40 gigabits per second—the fastest research network on the planet.*

The Montage TeraGrid portal is using these high performance resources to construct image mosaics. Users of the portal need only have a desktop computer running any standard web browser.

2 High Level Design

The Montage TeraGrid service accepts requests from two portals, one at JPL and one at ISI, underpinned by a common, distributed architecture. Figures 1 and 2 shows this common architecture. First we describe the JPL portal, shown in Figure 1. This portal is a prototype of one we will ultimately deploy for astronomers, who will submit mosaic

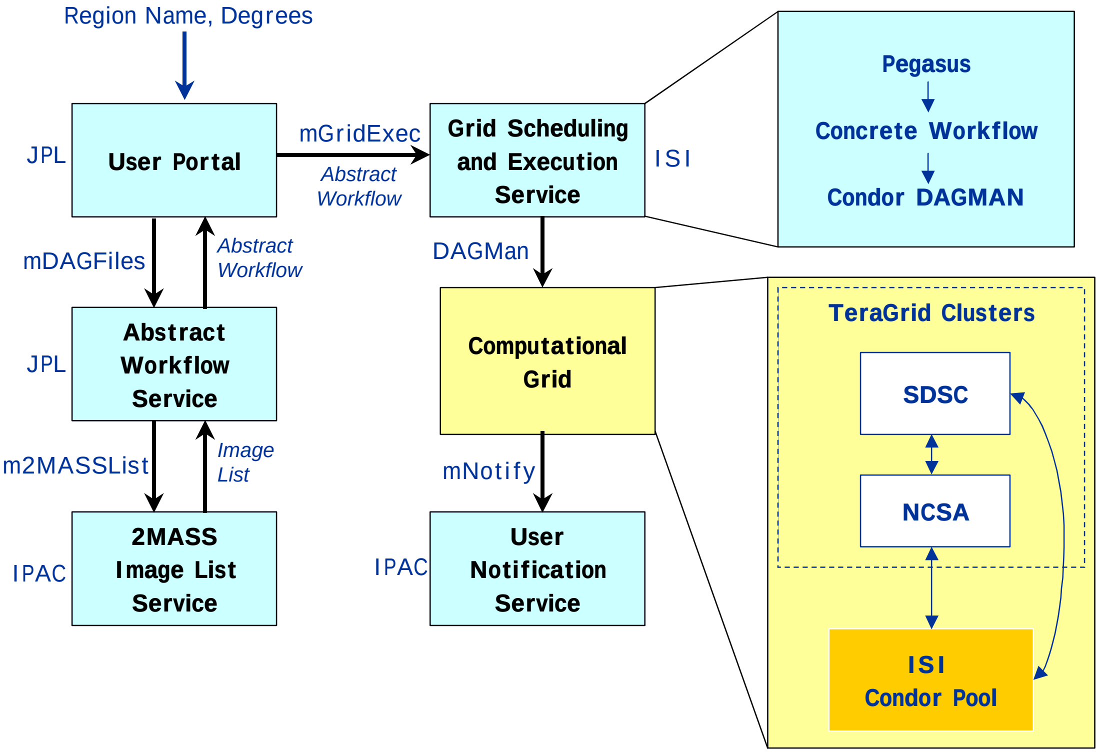


Figure 1. The distributed architecture of the Montage TeraGrid Portal.

requests through a simple web form that inputs parameters describing the mosaic (location on the sky, size, coordinate system, projection, etc). A service at JPL/Caltech is contacted to generate an abstract workflow, which specifies the processing jobs to be executed, input, output, and intermediate files to be read or written during the processing, and dependencies between the jobs. A 2MASS image list service at IPAC/Caltech is contacted to generate a list of the 2MASS images required to fulfill the mosaic request. The abstract workflow is passed to a service at the Information Sciences Institute (ISI), University of Southern California, which runs software called Pegasus (http://www.isi.edu/~deelman/pegasus.htm) to schedule the workflow on the TeraGrid. The resulting "concrete workflow" includes information about specific file locations on the grid and specific grid computers to be used for the processing. The workflow is then executed on the remote TeraGrid clusters using Condor DAGMan. DAGMan is a scheduler that submits jobs to Condor in an order specified by the concrete workflow. Condor queues the jobs for execution on the TeraGrid. More information on Condor and DAGMan can be found on the Condor web site at http://www.cs.wisc.edu/condor/. The last step in the mosaic processing is to contact a user notification service at IPAC/Caltech, which currently simply sends an email to the user with the URL of the Montage output.

The Montage grid portal is comprised of the following five main components, each having a client and server code:

1. User Portal
2. Abstract Workflow Service

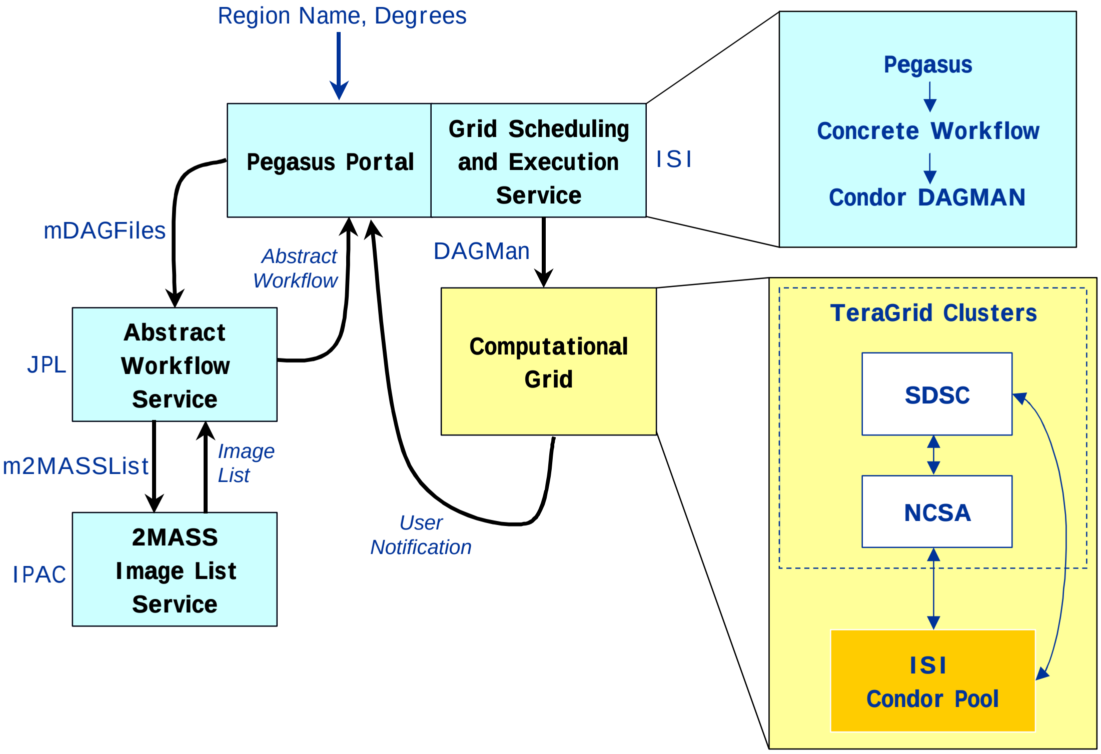


Figure 2. The preliminary design of the Montage TeraGrid Portal uses the Pegasus portal for user input and notification.

3. 2MASS Image List Service
4. Grid Scheduling and Execution Service
5. User Notification Service

The second portal is the Pegasus portal at ISI simply takes the place of the User Portal and the User Notification Service. This portal provides complete diagnostic and status information on the processing, and returns all intermediate products. Astronomers simply wishing to receive a mosaic would find the JPL portal more useful.

2.1 User Portal

The client code for the user portal is the ubiquitous web browser. Users fill out a simple web form with parameters that describe the mosaic to be constructed, including an object name or location, mosaic size, coordinate system, projection, and spatial sampling. Figure 3 shows a screen capture of the web form interface accessible at http://montage.jpl.nasa.gov/. The data in the web form are submitted to the CGI program using the HTTP `POST` method.

The server side of the user portal includes two main codes, both implemented as Perl scripts: `montage-cgi` and `montaged`. The `montage-cgi` program is a CGI script that is run by the Apache web server after the user presses the “Submit” button on the web form. This CGI script checks for validity of the request parameters, and stores the validated requests to disk for later processing. The `montaged` program has no direct connection to the web server and runs continuously as a daemon to process incoming mosaic requests. The processing for a request is done in two main steps:

1. Call the abstract workflow service client code
2. Call the grid scheduling and execution service client code and pass to it the output from the abstract workflow service client code

2.2 Abstract Workflow Service

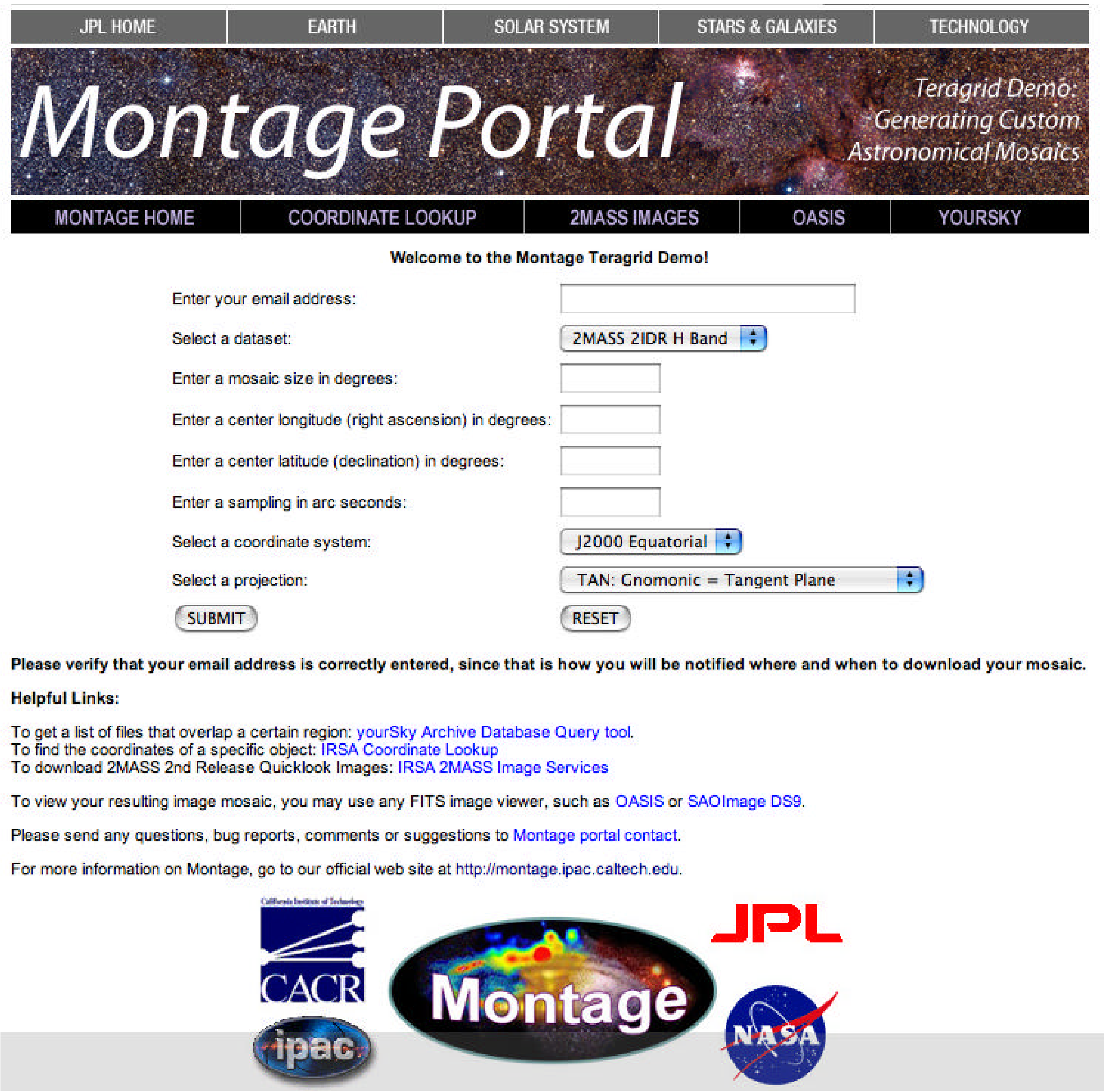
The client code for the abstract workflow service is mDAGFiles, a compiled ANSI C code, called with the following usage syntax:

```
mDAGFiles object|location size suffix zipfile
```

The input arguments are an object name or location on the sky (which must be specified as a single argument string), a mosaic size, a filename suffix, and an output zip archive filename. These input parameters are sent to the abstract workflow server using the HTTP `POST` method.

The server is a CGI perl script called `nph-mdag-cgi`, which creates a number of files, packs them into a zip archive file, and sends the zip file back to the calling client. The following files are included in the zip archive, with the specified filename if the `suffix` argument to `mDAGFiles` is specified as "`_SUF`" (Appendices A-G give sample files for a small mosaic of M51 built from just two images):

1. `adag_SUF.xml`: The abstract workflow as a directed acyclic graph (DAG) in XML; specifies the jobs and files to be encountered during the mosaic processing, and the dependencies between the jobs (see example in Appendix A)
2. `images_SUF.xml`: A table containing filenames and other attributes associated with the images needed to construct the mosaic (see example in Appendix B)



Welcome to the Montage Teragrid Demo!

Enter your email address:

Select a dataset: 2MASS 2IDR H Band

Enter a mosaic size in degrees:

Enter a center longitude (right ascension) in degrees:

Enter a center latitude (declination) in degrees:

Enter a sampling in arc seconds:

Select a coordinate system: J2000 Equatorial

Select a projection: TAN: Gnomonic = Tangent Plane

Please verify that your email address is correctly entered, since that is how you will be notified where and when to download your mosaic.

Helpful Links:

To get a list of files that overlap a certain region: yourSky Archive Database Query tool.
To find the coordinates of a specific object: IRSA Coordinate Lookup
To download 2MASS 2nd Release Quicklook Images: IRSA 2MASS Image Services

To view your resulting image mosaic, you may use any FITS image viewer, such as OASIS or SAOImage DS9.

Please send any questions, bug reports, comments or suggestions to Montage portal contact.

For more information on Montage, go to our official web site at http://montage.ipac.caltech.edu.

Figure 3. Montage grid portal web form interface.

3. `pimages_SUF.xml`: A table containing the filenames to be used for the images that have been reprojected by `mProject` (see example in Appendix C)
4. `cimages_SUF.xml`: A table containing the filenames to be used for the images that have been background corrected by `mBackground` (see example in Appendix D)
5. `fit_list_SUF.tbl`: A table containing the filenames output by `mFitplane` for the difference image fit plane parameters (see example in Appendix E)
6. `template_SUF.hdr`: Montage template header file describing output mosaic (see example in Appendix F)
7. `log_SUF.txt`: A log file with time stamps for each part of the processing (see example in Appendix G).

All of these files are required by the Grid Scheduling and Execution Service, described below). The abstract workflow in `adag_SUF.xml` specifies the filenames to be encountered and jobs to be run during the mosaic processing, and dependencies between jobs, which dictates which jobs can be run in parallel. A pictorial representation of an abstract workflow for a mosaic with three input images is shown in Figure 4 The `images_SUF.xml` file is created by querying the 2MASS Image List Service, described below. The `pimages_SUF.xml` and `cimages_SUF.xml` are required as input to the `mBgModel` and `mAdd` programs, respectively, and are created by simple text manipulation of the `images_SUF.xml` file. The `fit_list_SUF.tbl` file is

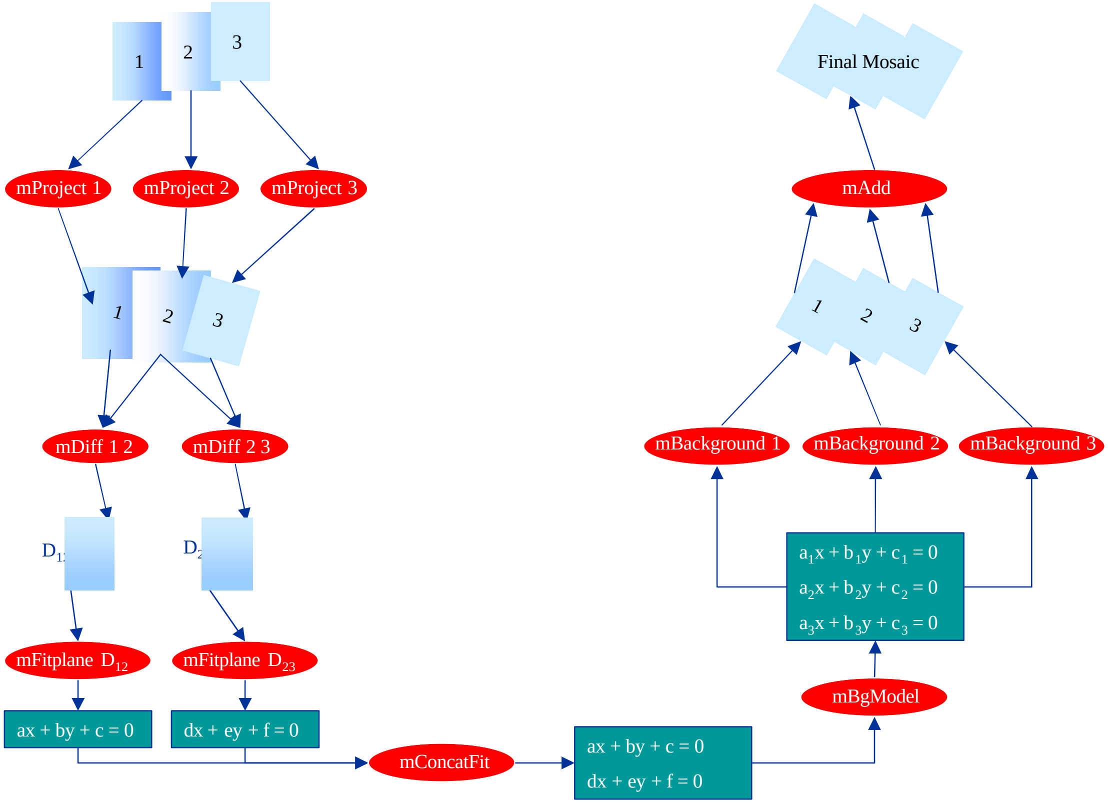


Figure 4. Example abstract workflow.

required as input to `mConcatFit` for merging of the individual fit plane files into one file required by `mBgModel`.

2.3 2MASS Image List Service

The 2MASS Image List Service is accessed via a client code called `m2MASSList`, which is called with the following user syntax:

```
m2MASSList object|location size outfile
```

The input arguments are an object name or location on the sky (which must be specified as a single argument string), a mosaic size in degrees, and an output file name. The 2MASS images that intersect the specified location on the sky are returned in `outfile` in a table, with columns that include the filenames and other attributes associated with the images.

2.4 Grid Scheduling and Execution Service

The Grid Scheduling and Execution Service is triggered using a simple client code called `mGridExec,` with the following calling syntax:

```
mGridExec zipfile
```

The input argument, `zipfile`, is the zip archive generated with the Abstract Workflow Service, described above.

On the server side, the user is authenticated on the Grid, and the work is first scheduled on the Grid using a program called Pegasus, and then executed using Condor DAGMan.

Pegasus is a workflow management system designed to map abstract workflows onto the Grid resources to produce concrete (executable) workflows. Pegasus consults various Grid information services, such as the Globus Monitoring and Discovery Service (MDS), the Globus Replica Location Service (RLS), the Metadata Catalog Service (MCS), and the Transformation Catalog to determine the available resources and data. Pegasus reduces the abstract workflow based on the available data. For example, if intermediate workflow products are registered in the RLS, Pegasus does not perform the transformations necessary to produce these products. The executable workflow generated by Pegasus identifies the resources where the computation will take place, the data movement for staging data in and out of the computation, and registers the newly derived data products in the RLS and MCS.

Users are authenticated on the TeraGrid using their Grid security credentials. The user first needs to save their proxy credential in the MyProxy server. MyProxy is a credential repository for the Grid that allows a trusted server (like our Grid Scheduling and Execution Service) to access grid credentials on the users behalf. This allows these credentials to be retrieved by the portal using the user's username and password. Once authentication is completed, Pegasus schedules the Montage workflow onto the TeraGrid or other clusters managed by PBS and Condor. The workflow is then submitted to

Condor DAGMan for execution. Upon completion, the final mosaic is delivered to a user-specified location and the User Notification Service, described below, is contacted.

2.5 User Notification Service

The last step in the grid processing is to notify the user with the URL where the mosaic may be downloaded. This notification is done by a remote user notification service at Caltech IPAC so that a new notification mechanism can be used later without having to modify the Grid Scheduling and Execution Service. Currently the user notification is done with a simple email, but a later version will use the Request Object Management Environment (ROME), being developed separately for the National Virtual Observatory. ROME will extend our portal with more sophisticated job monitoring, query and notification capabilities.

The User Notification Service is accessed with a simply client code, `mNotify`, with the following usage syntax:

```
mNotify jobID userID resultsURL
```

3 Modifications from Montage_v1.7.1

The Montage Grid portal implementation required adding a number of extra codes that were not included in the 1st release and modifying the usage syntax for a number of the codes. The usage syntax modifications were required because Pegasus determines some dependencies by direct filename matching. Every Montage module had to output at least one file as output to signal completion. Some of the Montage modules had an optional status argument (`-s status`) added for this purpose. Also, because of the way dependencies are handled by Pegasus, if module 2 depends on module 1, module 1 had to output a file that was read by module 2. These changes are summarized here:

1. Added `mDAGFiles` client program to access the Abstract Workflow Service:

   ```
   mDAGFiles object|location size suffix zipfile
   ```

2. Added `m2MASSList` client program to access the 2MASS Image List Service:

   ```
   m2MASSList object|location size outfile
   ```

3. Added `mGridExec` client program to access the Grid Scheduling and Execution Service:

   ```
   mGridExec zipfile
   ```

4. Added `mDAGTbls` program to take an image list from `m2MASSList` and the Montage template header file and produce two additional image lists, one for the projected image files and one for the background corrected image files:

   ```
   mDAGTbls [-d][–s status] images.tbl hdr.template
            projected.tbl corrected.tbl
   ```

5. Added mConcatFit program to concatenate the output from multiple mFitplane jobs. This program (not shown for simplicity on the archiecture diagrams) was needed because on the Grid, each mFitplane job could be run on a different cluster pool. The mBgModel code requires the fit plane parameters in a single file so mConcatFit is used to concatenate the individual mFitplane output files into a single file for mBgModel. The calling syntax is as follows:

```
mConcatFit [-d][-s status] statfiles.tbl fits.tbl statdir
```

6. Changed mFitplane calling syntax to store the output in a file rather than sending it to stdout. The status file (specified with –s outfile) is used to store the output. The new calling syntax is:

```
mFitplane [-b border] [-d level] [-s outfile] in.fits
```

7. Changed mBackground calling syntax to read the correction plane parameters from a file rather than from the command line. The old calling syntax had the plane parameters on the command line as A, B, C. These are replaced by two parameters: the image table output by mDAGTbls, and a single file containing a table of the correction parameters for each image in the image list. It is assumed that both of these tables include a column with the file number and these file numbers have a correspondance across the two tables. The new calling syntax is:

```
mBackground [-t] [-d level] in.fits out.fits images.tbl
                        corrfile.tbl
```

4 TeraGrid Performance and Issues

We have run the Pegasus-enabled Montage on a variety of resources: Condor pools, LSF- and PBS-managed clusters, and the TeraGrid (through PBS).

The following table summarizes the results of running a 2-degree M16 mosaic on the NCSA TeraGrid cluster. The total runtime of the workflow was 107 minutes. The workflow contained 1515 individual jobs. Below is a table that summarizes the runtimes of the individual workflow components.

number of jobs	job name	average runtime
1	mAdd	94.00 seconds
180	mBackground	2.64 seconds
1	mBgModel	11 seconds
1	mConcatFit	9 seconds
482	mDiff	2.89 seconds
483	mFitplane	2.55 seconds
180	mProject	130.52 seconds
183	transfer of data in	In the range of 5-30 seconds each

1	transfer of mosaic out	18: 03 minutes

To this point, our main goal was to demonstrate feasibility of running the Montage workflow in an automated fashion on the TeraGrid with some amount of performance improvement over the sequential version. Currently, Pegasus schedules the workflow as a set of small jobs (as seen in the table above, some of these jobs run only a few seconds.) Obviously scheduling too many little jobs suffers from large overheads. In fact, if this processing was run on a single TeraGrid processor, it would have taken 445 minutes, so we are not taking very much advantage of the TeraGrid's parallelism. However, initially structuring the workflow that way allows us to expose the highest degree of parallelism.

For comparison, we previously (yourSky Baseline Performance document) reported mosiacking a 1-square degree region of 2MASS data with yourSky on 4 194 MHz MIPS R10000 processors of an SGI PowerOnyx in 428.4 seconds. As this machine has been retired, we can compare this to yourSky running on 4 600 MHz MIPS R14000 processors, where we have mosaicked a 4-square degree 2MASS image in 9 minutes. This is reasonable, since the work involved is 4 times more, and the processors are about 3 times faster. Computing the same 4-square degree mosaic using Montage 1.7 with handscripted parallelization takes 153 minutes, making this version of Montage roughly 17 times slower than yourSky.

There are two issues brought out in this performance discussion. First, the current version of Montage is slow when compared with yourSky. This is because we were more concerned with accuracy (preservation of calibration and astrometric fidelity) than performance in this version of Montage, and we will be optimizing future versions of Montage to increase the performance without reducing the accuracy. That is, the performance figures reflect a computational burden that must be borne in delivering science grade products. Second, the use of our current Grid software with our current DAG causes the parallel performance on the TeraGrid to be sub-optimal. We plan to address this in three ways: making Pegasus aggregate nodes in the workflow in a way that would reduce the overheads for given target systems; encouraging the Condor developers to reduce the per-job overhead; and examining alternate methods for distributing the work on the Grid, such as something similar to the handscripting on the MIPS machines. Each option has advantages and disadvantages that will be weighed as we go forward.

Appendix A: Sample XML Abstract Workflow for M51

adag_M51.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<adag xmlns="http://www.griphyn.org/chimera/DAX"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.griphyn.org/chimera/DAX
http://www.griphyn.org/chimera/dax-1.5.xsd"
```

```
      count="1" index="0" name="test">

     <filename file="2mass-atlas-980527n-j0230033.fits"
link="input" isTemporary="false"/>
     <filename file="2mass-atlas-980527n-j0240232.fits"
link="input" isTemporary="false"/>

     <filename file="p2mass-atlas-980527n-j0230033.fits"
link="inout" isTemporary="true" temporaryHint="temp"/>
     <filename file="p2mass-atlas-980527n-j0240232.fits"
link="inout" isTemporary="true" temporaryHint="temp"/>

     <filename file="diff.1.2.fits" link="inout"
isTemporary="true" temporaryHint="temp"/>

     <filename file="fit.1.2.txt" link="inout"
isTemporary="true" temporaryHint="temp"/>
     <filename file="fits_M51.tbl" link="inout"
isTemporary="true" temporaryHint="temp"/>

     <filename file="corrections.tbl" link="inout"
isTemporary="true" temporaryHint="temp"/>

     <filename file="c2mass-atlas-980527n-j0230033.fits"
link="inout" isTemporary="true" temporaryHint="temp"/>
     <filename file="c2mass-atlas-980527n-j0240232.fits"
link="inout" isTemporary="true" temporaryHint="temp"/>

     <filename file="p2mass-atlas-980527n-
j0230033_area.fits" link="inout" isTemporary="true"
temporaryHint="temp"/>
     <filename file="c2mass-atlas-980527n-
j0230033_area.fits" link="inout" isTemporary="true"
temporaryHint="temp"/>
     <filename file="p2mass-atlas-980527n-
j0240232_area.fits" link="inout" isTemporary="true"
temporaryHint="temp"/>
     <filename file="c2mass-atlas-980527n-
j0240232_area.fits" link="inout" isTemporary="true"
temporaryHint="temp"/>
     <filename file="diff.1.2_area.fits" link="output"
isTemporary="true" temporaryHint="temp"/>

     <filename file="pimages_M51.tbl" link="input"
isTemporary="false"/>
     <filename file="cimages_M51.tbl" link="input"
isTemporary="false"/>
```

```
     <filename file="fit_list_M51.tbl" link="input"
isTemporary="false"/>
     <filename file="template_M51.hdr" link="input"
isTemporary="false"/>
     <filename file="out_M51.fits" link="output"
isTemporary="false"/>

     <filename file="out_M51_area.fits" link="output"
isTemporary="false"/>

     <job name="mProject" id="ID000001">
          <argument>
               <filename file="2mass-atlas-980527n-
j0230033.fits" link="input" isTemporary="false"/>
               <filename file="p2mass-atlas-980527n-
j0230033.fits" link="output" isTemporary="true"/>
               <filename file="template_M51.hdr"
link="input" isTemporary="false"/>
          </argument>
          <uses file="2mass-atlas-980527n-j0230033.fits"
link="input" isTemporary="false"/>
          <uses file="p2mass-atlas-980527n-j0230033.fits"
link="output" isTemporary="true"/>
          <uses file="p2mass-atlas-980527n-
j0230033_area.fits" link="output" isTemporary="true"/>
          <uses file="template_M51.hdr" link="input"
isTemporary="false"/>
     </job>

     <job name="mProject" id="ID000002">
          <argument>
               <filename file="2mass-atlas-980527n-
j0240232.fits" link="input" isTemporary="false"/>
               <filename file="p2mass-atlas-980527n-
j0240232.fits" link="output" isTemporary="true"/>
               <filename file="template_M51.hdr"
link="input" isTemporary="false"/>
          </argument>
          <uses file="2mass-atlas-980527n-j0240232.fits"
link="input" isTemporary="false"/>
          <uses file="p2mass-atlas-980527n-j0240232.fits"
link="output" isTemporary="true"/>
          <uses file="p2mass-atlas-980527n-
j0240232_area.fits" link="output" isTemporary="true"/>
          <uses file="template_M51.hdr" link="input"
isTemporary="false"/>
     </job>
     <job name="mDiff" id="ID000003">
          <argument>
               <filename file="p2mass-atlas-980527n-
j0230033.fits" link="input" isTemporary="true"/>
               <filename file="p2mass-atlas-980527n-
j0240232.fits" link="input" isTemporary="true"/>
               <filename file="diff.1.2.fits" link="output"
isTemporary="true"/>
               <filename file="template_M51.hdr"
link="input" isTemporary="false"/>
          </argument>
          <uses file="p2mass-atlas-980527n-j0230033.fits"
link="input" isTemporary="true"/>
          <uses file="p2mass-atlas-980527n-
j0230033_area.fits" link="input" isTemporary="true"/>
          <uses file="p2mass-atlas-980527n-j0240232.fits"
link="input" isTemporary="true"/>
          <uses file="p2mass-atlas-980527n-
j0240232_area.fits" link="input" isTemporary="true"/>
          <uses file="diff.1.2.fits" link="output"
isTemporary="true"/>
          <uses file="diff.1.2_area.fits" link="output"
isTemporary="true"/>
          <uses file="template_M51.hdr" link="input"
isTemporary="false"/>
     </job>

     <job name="mFitplane" id="ID000004">
          <argument>
               -s
               <filename file="fit.1.2.txt" link="output"
isTemporary="true"/>
               <filename file="diff.1.2.fits" link="input"
isTemporary="true"/>
          </argument>
          <uses file="fit.1.2.txt" link="output"
isTemporary="true"/>
          <uses file="diff.1.2.fits" link="input"
isTemporary="true"/>
     </job>

     <job name="mConcatFit" id="ID000005">
          <argument>
               <filename file="fit_list_M51.tbl"
link="input" isTemporary="false"/>
```

```
               <filename file="fits_M51.tbl" link="output"
isTemporary="true"/>
               .
          </argument>
          <uses file="fit_list_M51.tbl" link="input"
isTemporary="false"/>
          <uses file="fits_M51.tbl" link="output"
isTemporary="true"/>
          <uses file="fit.1.2.txt" link="input"
isTemporary="true"/>
     </job>

     <job name="mBgModel" id="ID000006">
          <argument>
               <filename file="pimages_M51.tbl"
link="input" isTemporary="false"/>
               <filename file="fits_M51.tbl" link="input"
isTemporary="true"/>
               <filename file="corrections.tbl"
link="output" isTemporary="true"/>
          </argument>
          <uses file="pimages_M51.tbl" link="input"
isTemporary="false"/>
          <uses file="fits_M51.tbl" link="input"
isTemporary="true"/>
          <uses file="corrections.tbl" link="output"
isTemporary="true"/>
     </job>

     <job name="mBackground" id="ID000007">
          <argument>
               -t
               <filename file="p2mass-atlas-980527n-
j0230033.fits" link="input" isTemporary="true"/>
               <filename file="c2mass-atlas-980527n-
j0230033.fits" link="output" isTemporary="true"/>
               <filename file="pimages_M51.tbl"
link="input" isTemporary="false"/>
               <filename file="corrections.tbl"
link="input" isTemporary="true"/>
          </argument>
          <uses file="p2mass-atlas-980527n-j0230033.fits"
link="input" isTemporary="true"/>
          <uses file="p2mass-atlas-980527n-
j0230033_area.fits" link="input" isTemporary="true"/>
          <uses file="pimages_M51.tbl" link="input"
isTemporary="false"/>
```

```
          <uses file="corrections.tbl" link="input"
isTemporary="true"/>
          <uses file="c2mass-atlas-980527n-j0230033.fits"
link="output" isTemporary="true"/>
          <uses file="c2mass-atlas-980527n-
j0230033_area.fits" link="output" isTemporary="true"/>
     </job>

     <job name="mBackground" id="ID000008">
          <argument>
               -t
               <filename file="p2mass-atlas-980527n-
j0240232.fits" link="input" isTemporary="true"/>
               <filename file="c2mass-atlas-980527n-
j0240232.fits" link="output" isTemporary="true"/>
               <filename file="pimages_M51.tbl"
link="input" isTemporary="false"/>
               <filename file="corrections.tbl"
link="input" isTemporary="true"/>
          </argument>
          <uses file="p2mass-atlas-980527n-j0240232.fits"
link="input" isTemporary="true"/>
          <uses file="p2mass-atlas-980527n-
j0240232_area.fits" link="input" isTemporary="true"/>
          <uses file="pimages_M51.tbl" link="input"
isTemporary="false"/>
          <uses file="corrections.tbl" link="input"
isTemporary="true"/>
          <uses file="c2mass-atlas-980527n-j0240232.fits"
link="output" isTemporary="true"/>
          <uses file="c2mass-atlas-980527n-
j0240232_area.fits" link="output" isTemporary="true"/>
     </job>

     <job name="mAdd" id="ID000009">
          <argument>
               <filename file="cimages_M51.tbl"
link="input" isTemporary="false"/>
               <filename file="template_M51.hdr"
link="input" isTemporary="false"/>
               <filename file="out_M51.fits" link="output"
isTemporary="false"/>
          </argument>
          <uses file="cimages_M51.tbl" link="input"
isTemporary="false"/>
          <uses file="template_M51.hdr" link="input"
isTemporary="false"/>
```

```
          <uses file="out_M51.fits" link="output"
isTemporary="false"/>
          <uses file="out_M51_area.fits" link="output"
isTemporary="false"/>
          <uses file="c2mass-atlas-980527n-j0230033.fits"
link="input" isTemporary="true"/>
          <uses file="c2mass-atlas-980527n-
j0230033_area.fits" link="input" isTemporary="true"/>
          <uses file="c2mass-atlas-980527n-j0240232.fits"
link="input" isTemporary="true"/>
          <uses file="c2mass-atlas-980527n-
j0240232_area.fits" link="input" isTemporary="true"/>
     </job>

     <child ref="ID000003">
          <parent ref="ID000001"/>
          <parent ref="ID000002"/>
     </child>

     <child ref="ID000004">
          <parent ref="ID000003"/>
     </child>

     <child ref="ID000005">
          <parent ref="ID000004"/>
     </child>

     <child ref="ID000006">
          <parent ref="ID000005"/>
     </child>

     <child ref="ID000007">
          <parent ref="ID000001"/>
          <parent ref="ID000006"/>
     </child>

     <child ref="ID000008">
          <parent ref="ID000002"/>
          <parent ref="ID000006"/>
     </child>

     <child ref="ID000009">
          <parent ref="ID000007"/>
          <parent ref="ID000008"/>
     </child>

</adag>
```

Appendix B: Sample Image Table

images_M51.tbl:

```
\datatype=fitshdr
| cntr|  ctype1|  ctype2|naxis1|naxis2|    crval1|
crval2| crpix1| crpix2|     cdelt1|     cdelt2|  crota2|
SRB|                                          Quicklook|
file|
|  int|    char|    char|   int|   int|    double|
double| double| double|     double|     double|  double|
char|                                               char|
char|
     1 RA---SIN DEC--SIN    512   1024 202.361114
47.326477  256.50  512.50 -0.00027778  0.00027778 -0.02065
/home/thh.caltech/n-213031332-980527-023-0033-j
\2MASSDataPath\/980527n/s023/image/ji0230033.fits.H 2mass-
atlas-980527n-j0230033.fits
     2 RA---SIN DEC--SIN    512   1024 202.521394
47.113192  256.50  512.50 -0.00027778  0.00027778  0.00339
/home/thh.caltech/n-213031332-980527-024-0232-j
\2MASSDataPath\/980527n/s024/image/ji0240232.fits.H 2mass-
atlas-980527n-j0240232.fits
```

Appendix C: Sample Projected Image Table

pimages_M51.tbl:

```
\datatype=fitshdr
| cntr|  ctype1|  ctype2|naxis1|naxis2|    crval1|
crval2|    crpix1|    crpix2|     cdelt1|     cdelt2|
crota2|equinox|                               file|
|  int|    char|    char|   int|   int|    double|
double|    double|    double|     double|     double|
double|    int|                               char|
     1 RA---TAN DEC--TAN    516   1027 202.441157
47.219954      63.50     130.00 -0.00027778  0.00027778
0.05936    2000 p2mass-atlas-980527n-j0230033.fits
     2 RA---TAN DEC--TAN    513   1024 202.441157
47.219954     452.50     896.00 -0.00027778  0.00027778
0.05936    2000 p2mass-atlas-980527n-j0240232.fits
```

Appendix D: Sample Corrected Image Table

cimages_M51.tbl:

```
\datatype=fitshdr
| cntr|  ctype1|  ctype2|naxis1|naxis2|    crval1|
crval2|    crpix1|    crpix2|     cdelt1|     cdelt2|
crota2|equinox|                               file|
|  int|    char|    char|   int|   int|    double|
double|    double|    double|     double|     double|
double|    int|                               char|
     1 RA---TAN DEC--TAN    516   1027 202.441157
47.219954      63.50     130.00 -0.00027778  0.00027778
0.05936    2000 c2mass-atlas-980527n-j0230033.fits
     2 RA---TAN DEC--TAN    513   1024 202.441157
47.219954     452.50     896.00 -0.00027778  0.00027778
0.05936    2000 c2mass-atlas-980527n-j0240232.fits
```

Appendix E: Sample Fit Plane File Table

fit_list_M51.tbl:

```
|cntr1|cntr2|   stat    |
   1     2   fit.1.2.txt
```

Appendix F: Sample Montage Template File

template_M51.hdr:

```
SIMPLE  = T
BITPIX  = -64
NAXIS   = 2
NAXIS1  = 906
NAXIS2  = 1793
CTYPE1  = 'RA---TAN'
CTYPE2  = 'DEC--TAN'
EQUINOX = 2000
CRVAL1  =  202.441156834
CRVAL2  =   47.219953535
CDELT1  =   -0.000277780
CDELT2  =    0.000277780
CRPIX1  =       453.5000
CRPIX2  =       897.0000
CROTA2  =    0.059358733
END
```

Appendix G: Sample Abstract Workflow Service Log File

log_M51.txt:

```
Fri Jan  9 09:33:34 PST 2004 : Started DAG processing for
location "m51" and size ".2"
[struct stat="OK", count="3"]
Fri Jan  9 09:33:38 PST 2004 : Created input image table
with m2MASSList
[struct stat="OK", count=2]
Fri Jan  9 09:33:38 PST 2004 : Created Montage template
header file with mMakeHdr
[struct stat="OK", count="2"]
```

```
Fri Jan  9 09:33:38 PST 2004 : Created projected and
background corrected image tables with mDAGTbls
[struct stat="OK", count=1]
Fri Jan  9 09:33:38 PST 2004 : Created list of overlapping
images with mOverlaps
Fri Jan  9 09:33:38 PST 2004 : Finished DAG processing for
location "m51" and size ".2"
```