Pegasus: Mapping complex applications onto the Grid

Ewa Deelman

Center for Grid Technologies

USC Information Sciences Institute

Pegasus Acknowledgements

- Ewa Deelman, Carl Kesselman, Saurabh Khurana, Gaurang Mehta, Sonal Patil, Gurmeet Singh, Mei-Hui Su, Karan Vahi (Center for Grid Computing, ISI)
- James Blythe, Yolanda Gil (Intelligent Systems Division, ISI)
- http://pegasus.isi.edu
- Research funded as part of the NSF GriPhyN, NVO and SCEC projects.

Outline

- The GriPhyN project and Grid Applications
- Workflow Management in Grids
- Pegasus, Planning for Execution in Grids
  - Framework Description
  - Generation of Executable Workflows
- Applications Using Pegasus
- Future Research Directions

GriPhyN Data Grid Challenge

- Provide a framework that enables Virtual Organizations around the world to perform computationally demanding analysis of large, geographically distributed datasets.
- The Virtual Organizations are large and highly distributed
- The datasets are large, currently on the order of Terabytes and expected to grow to the level of 100s of Petabytes in the next decade
- Provide a seamless access to data: experimental raw data or processed data products
- Enable a user/application to ask for any domain-specific data, whether computed or not

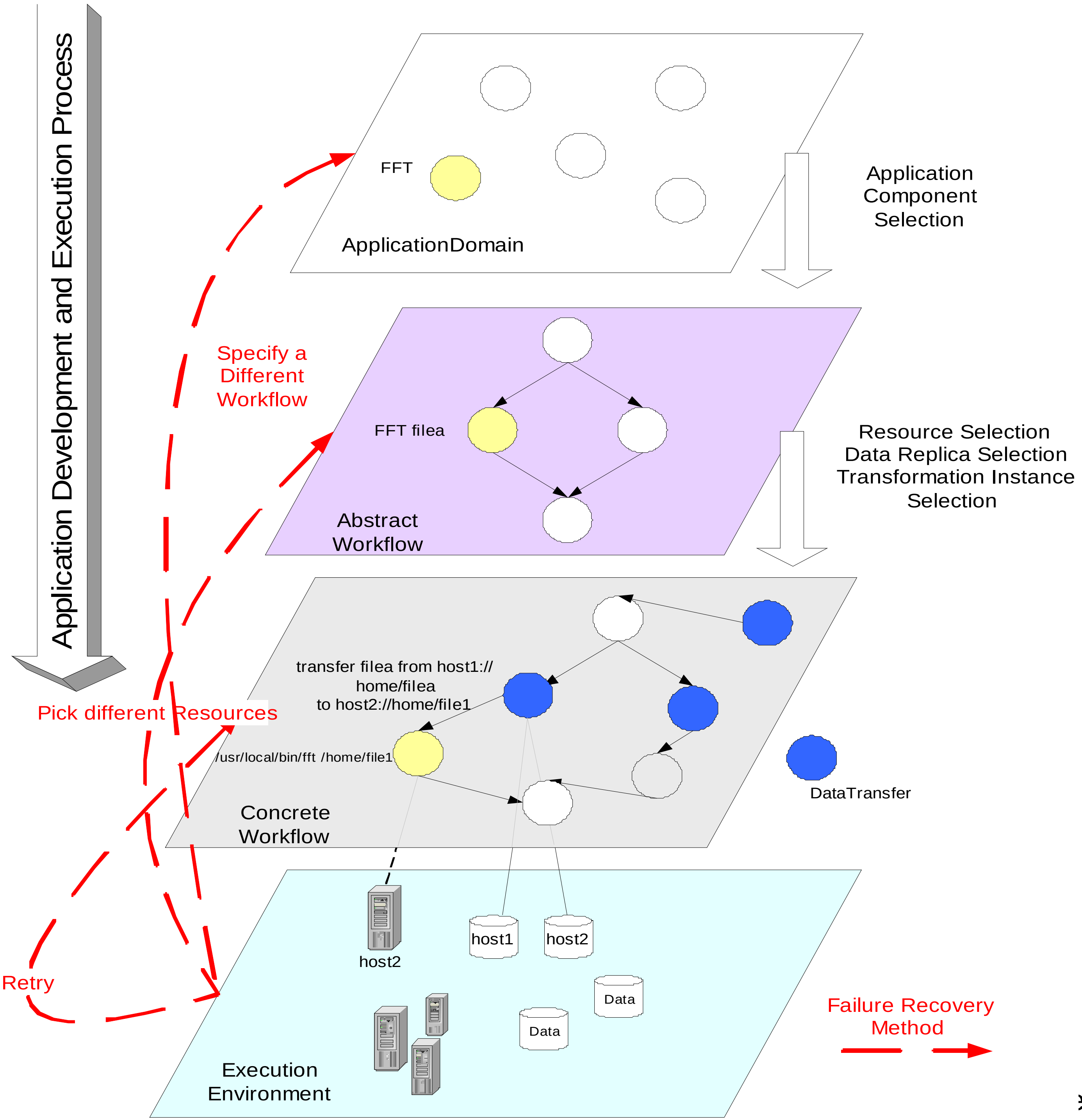
Concept of Virtual Data

Grid Applications

- Increasing in the level of complexity
- Use of individual application components
- Reuse of individual intermediate data products (files)
- Description of Data Products using Metadata Attributes

- Execution environment is complex and very dynamic
    - Resources come and go
    - Data is replicated
    - Components can be found at various locations or staged in on demand

- Separation between
    - the application description
    - the actual execution description



Application Development and Execution Process

FFT

ApplicationDomain

Application Component Selection

Specify a Different Workflow

FFT filea

Abstract Workflow

Resource Selection
Data Replica Selection
Transformation Instance Selection

transfer filea from host1://home/filea
to host2://home/file1

Pick different Resources

/usr/local/bin/fft /home/file1

DataTransfer

Concrete Workflow

host2

host1

host2

Data

Data

Retry

Execution Environment

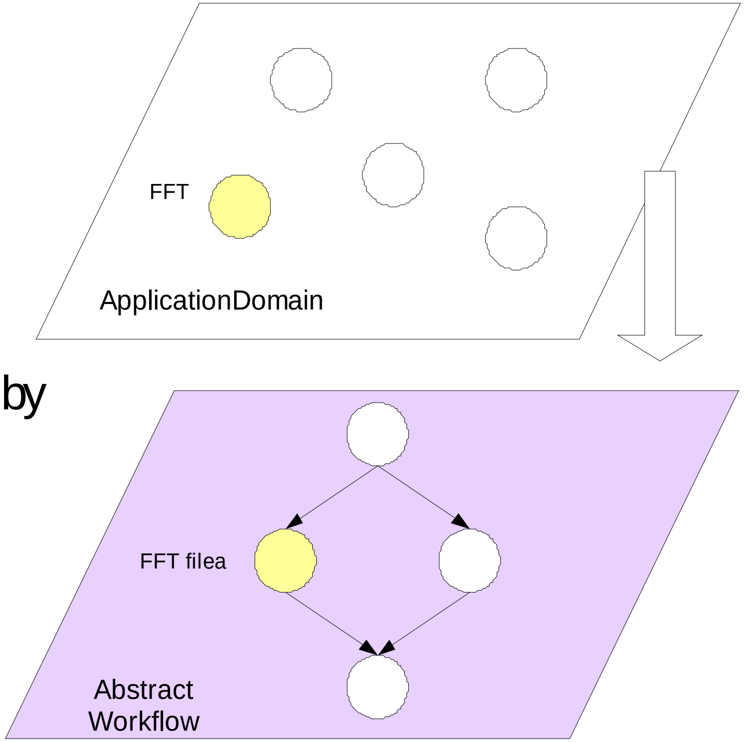
Failure Recovery Method

Generating an Abstract Workflow

- Available Information
  - Specification of component capabilities
  - Ability to generate the desired data products

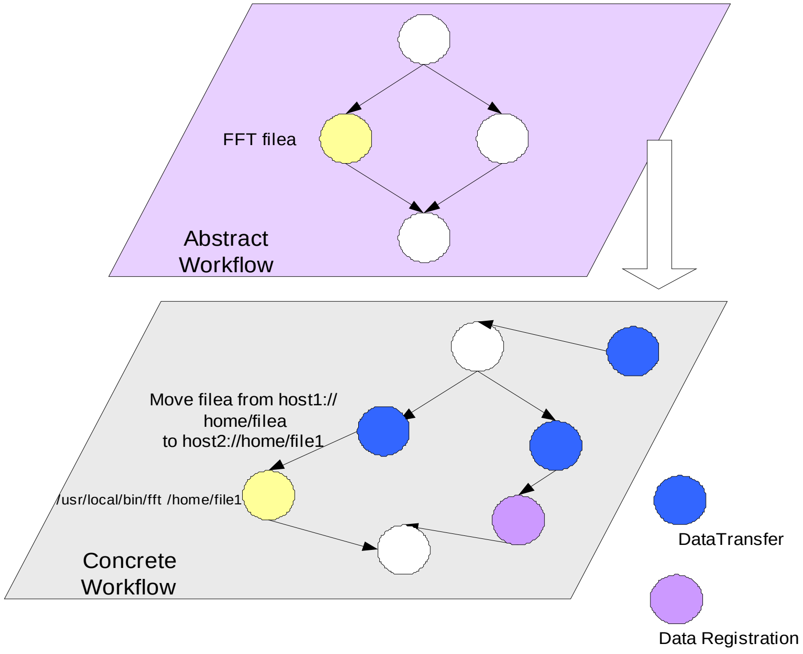
Select and configure application components to form an abstract workflow

  - assign input files that exist or that can be generated by other application components.
  - specify the order in which the components must be executed
  - components and files are referred to by their logical names
    - Logical transformation name
    - Logical file name
    - Both transformations and data can be replicated



Generating a Concrete Workflow

- Information
  - location of files and component Instances
  - State of the Grid resources
- Select specific_
  - Resources
  - Files
  - Add jobs required to form a concrete workflow that can be executed in the Grid environment
    - Data movement
  - Data registration
  - Each component in the abstract workflow is turned into an executable job

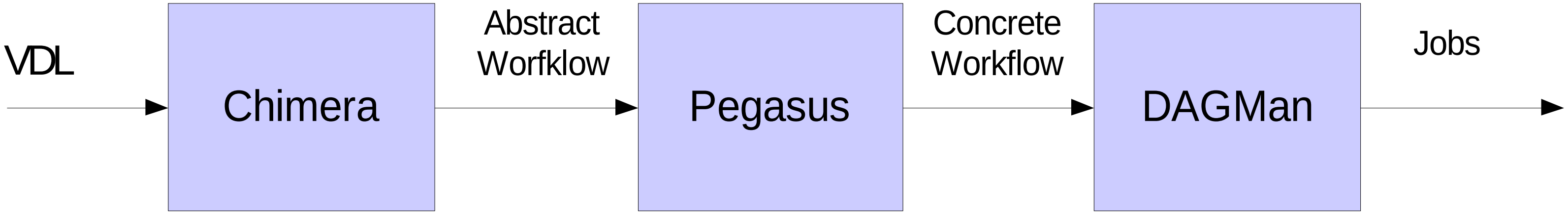


Why Automate Workflow Generation?

- **Usability:** **Limit User's necessary Grid knowledge**
    - Monitoring and Directory Service
    - Replica Location Service
- **Complexity:**
    - User needs to make choices
        - Alternative application components
        - Alternative files
        - Alternative locations
    - The user may reach a dead end
    - Many different interdependencies may occur among components
- **Solution cost:**
    - Evaluate the alternative solution costs
        - Performance
        - Reliability
        - Resource Usage
- **Global cost:**
    - minimizing cost within a community or a virtual organization
    - requires reasoning about individual user's choices in light of other user's choices

GriPhyN's
Executable Workflow Construction

- Build an abstract workflow based on VDL descriptions (Chimera)
- Build an executable workflow based on the abstract workflows (Pegasus)
- Execute the workflow (Condor's DAGMan)



Chimera: Creating Abstract Workflows

- Developed at ANL (Foster, Voeckler, Wilde)
- Chimera's Virtual Data Language (VDL) allows for the description of an abstract workflow
- Transformations:
  - general description of the transformation applied to data, use logical transformation name

```
TR          galMorph( in redshift, in pixScale, in zeroPoint, in Ho, in om, in flat,
                      in image, out galMorph ) {
                      …  }
```

Chimera :
Creating Abstract Workflows

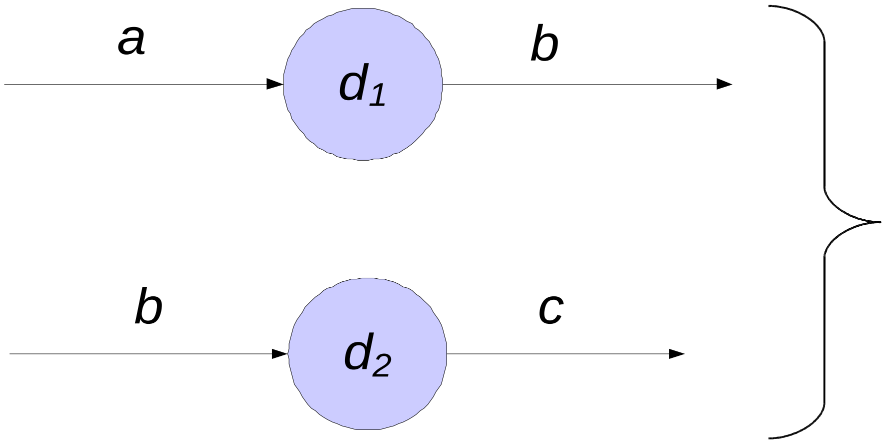
- Derivations are instantiations of TRs
  - Identify particular logical input and output file names
  - Identify actual parameters

```
DV d1->galMorph(
        redshift="0.027886",
        image=@{in:"NGP9_F323-0927589.fit"},
        pixScale="2.8319331070350 62E-4",
        zeroPoint="0",
        Ho="100",
        om="0.3",
        flat="1",
        galMorph=@{out:"NGP9_F323-0927589.txt"}  );
```

Abstract Workflow Generation

- Definitions for transformations and derivations are stored in Chimera’s Database
- Database can be browsed
- User queries Chimera giving it a logical filename

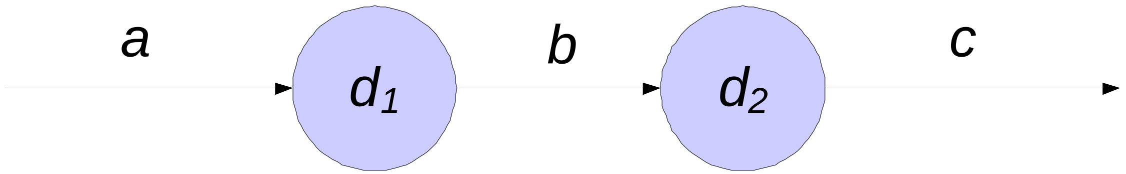
VDL and Abstract Workflow



VDL descriptions

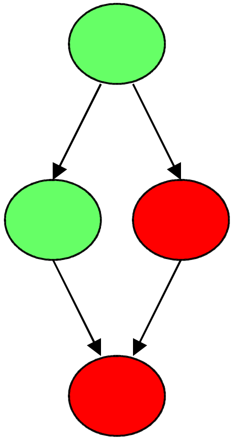
User request data file “c”

Abstract Workflow



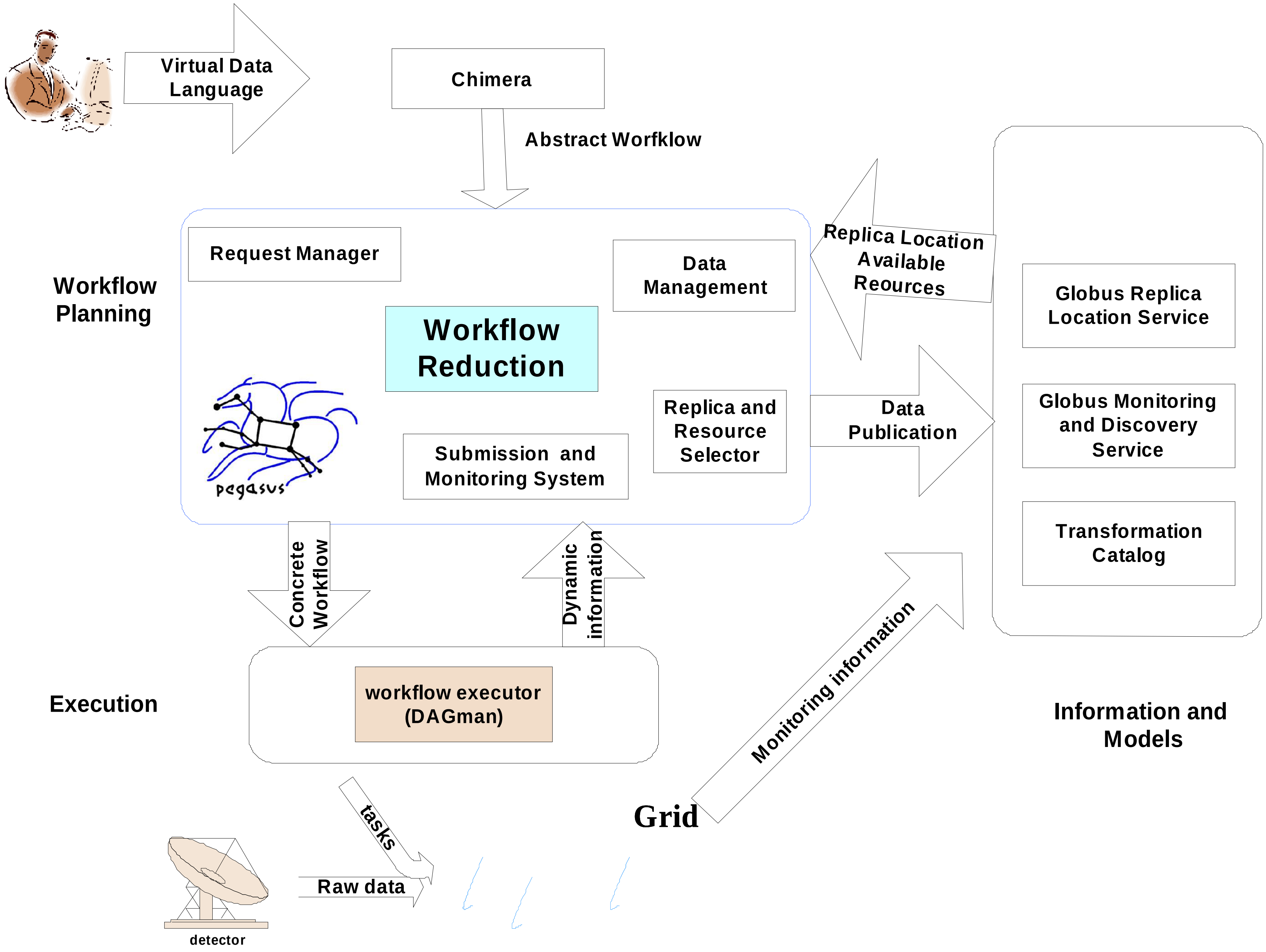
Condor’s DAGMan

- Developed at UW Madison (Livny)
- Executes a concrete workflow
- Makes sure the dependencies are followed
- Execute the jobs specified in the workflow
  - Execution
  - Data movement
  - Catalog updates
- Provides a “rescue DAG” in case of failure



Pegasus:
Planning for Execution in Grids

- Maps from abstract to concrete workflow
  - Algorithmic and AI-based techniques
- Automatically locates physical locations for both components (transformations) and data
- Finds appropriate resources to execute
- Reuses existing data products where applicable
- Publishes newly derived data products
  - Chimera virtual data catalog
  - Provides provenance information



Virtual Data Language

Chimera

Abstract Worfklow

Workflow Planning

Request Manager

Data Management

Workflow Reduction

Replica Location Available Reources

Pegasus

Submission and Monitoring System

Replica and Resource Selector

Data Publication

Globus Replica Location Service

Globus Monitoring and Discovery Service

Transformation Catalog

Concrete Workflow

Dynamic information

Execution

workflow executor (DAGman)

Monitoring information

Information and Models

Grid

tasks

Raw data

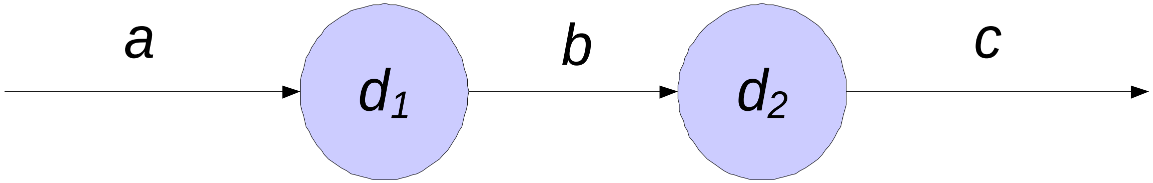
detector

Information Components Used by Pegasus

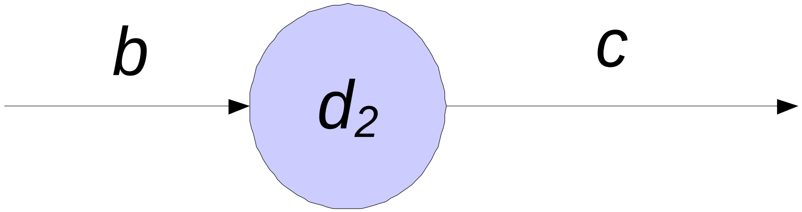
- Globus Monitoring and Discovery Service (MDS)
  - Locates available resources
  - Finds resource properties
    - Dynamic: load, queue length
    - Static: location of gridftp server, RLS, etc
- Globus Replica Location Service
  - Locates data that may be replicated
  - Registers new data products
- Transformation Catalog
  - Locates installed executables

Example Workflow Reduction

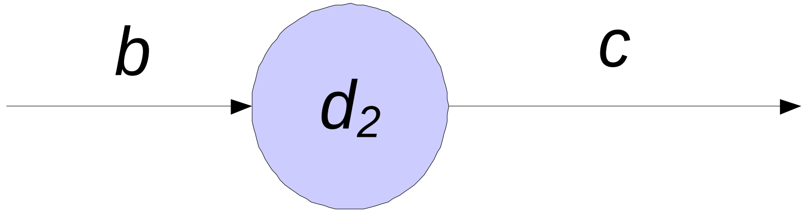
- Original abstract workflow



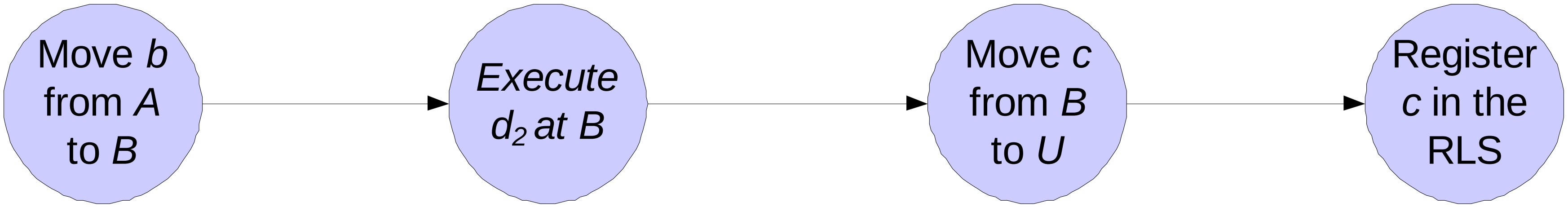
- If “b” already exists (as determined by query to the RLS), the workflow can be reduced



Mapping from abstract to concrete



- Query RLS, MDS, and TC, schedule computation and data movement



Applications Using Chimera, Pegasus and DAGMan

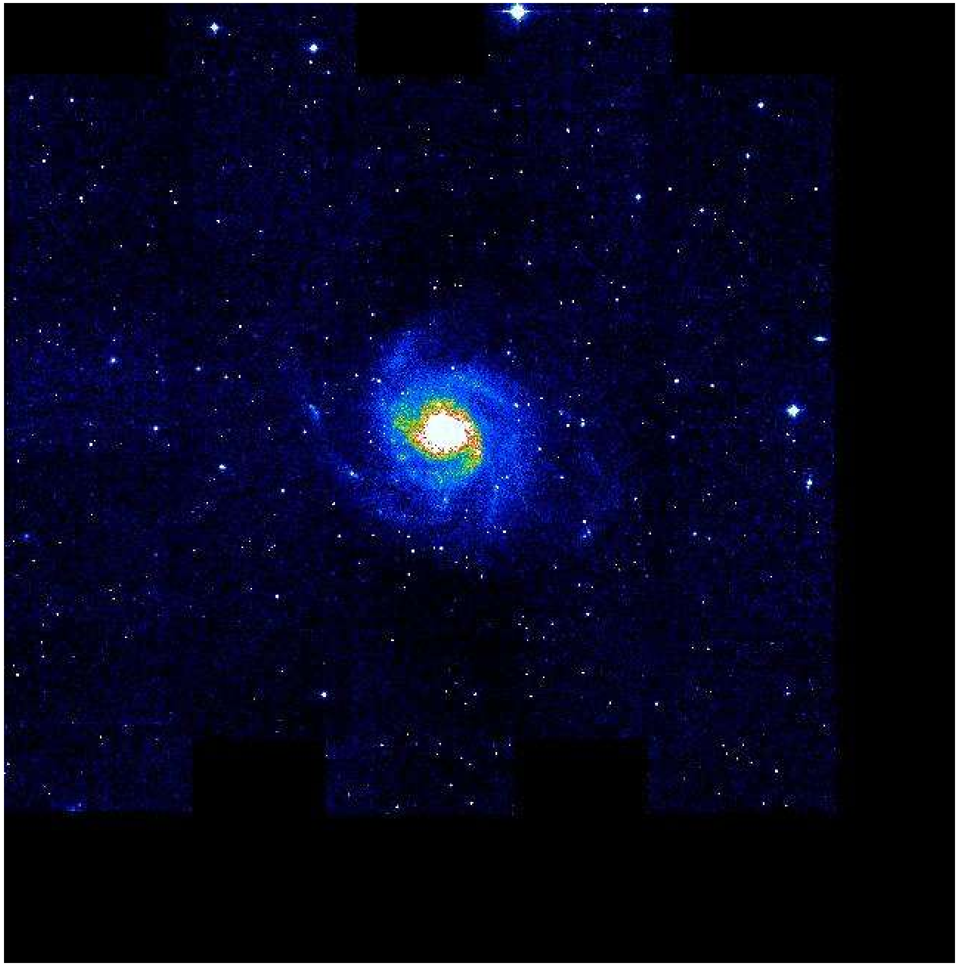
- GriPhyN applications:
  - High-energy physics: Atlas, CMS (many)
  - Astronomy: SDSS (Fermi Lab, ANL)
  - Gravitational-wave physics: LIGO (Caltech, UWM)
- Astronomy:
  - Galaxy Morphology (NCSA, JHU, Fermi, many others, NVO-funded)
- Biology
  - BLAST (ANL, PDQ-funded)
- Neuroscience
  - Tomography for Telescience(SDSC, NIH-funded)

Pegasus interfaces

- Main interface: command-line interface
- Applications can also be integrated with a portal environment
- Demonstrated the portal at SC 2003
  - LIGO-gravitational-wave physics
  - Montage-astronomy
- Much of the portal is application-independent

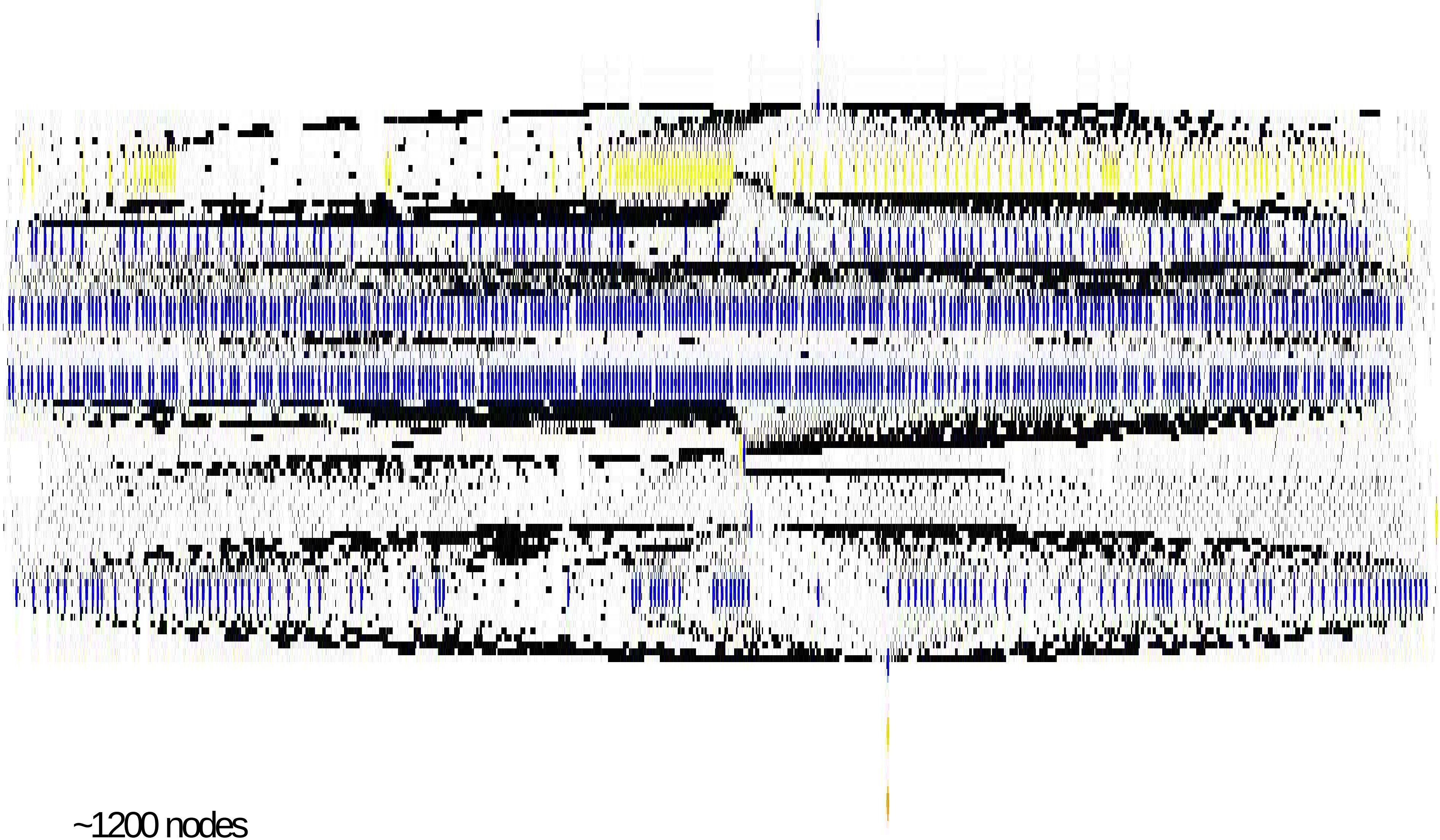
Montage

- Montage (NASA and NVO)
  - Deliver science-grade custom mosaics on demand
  - Produce mosaics from a wide range of data sources (possibly in different spectra)
  - User-specified parameters of projection, coordinates, size, rotation and spatial sampling.



Mosaic created by Pegasus based Montage from a run of the M101 galaxy images on the Teragrid.

Small Montage Workflow



~1200 nodes

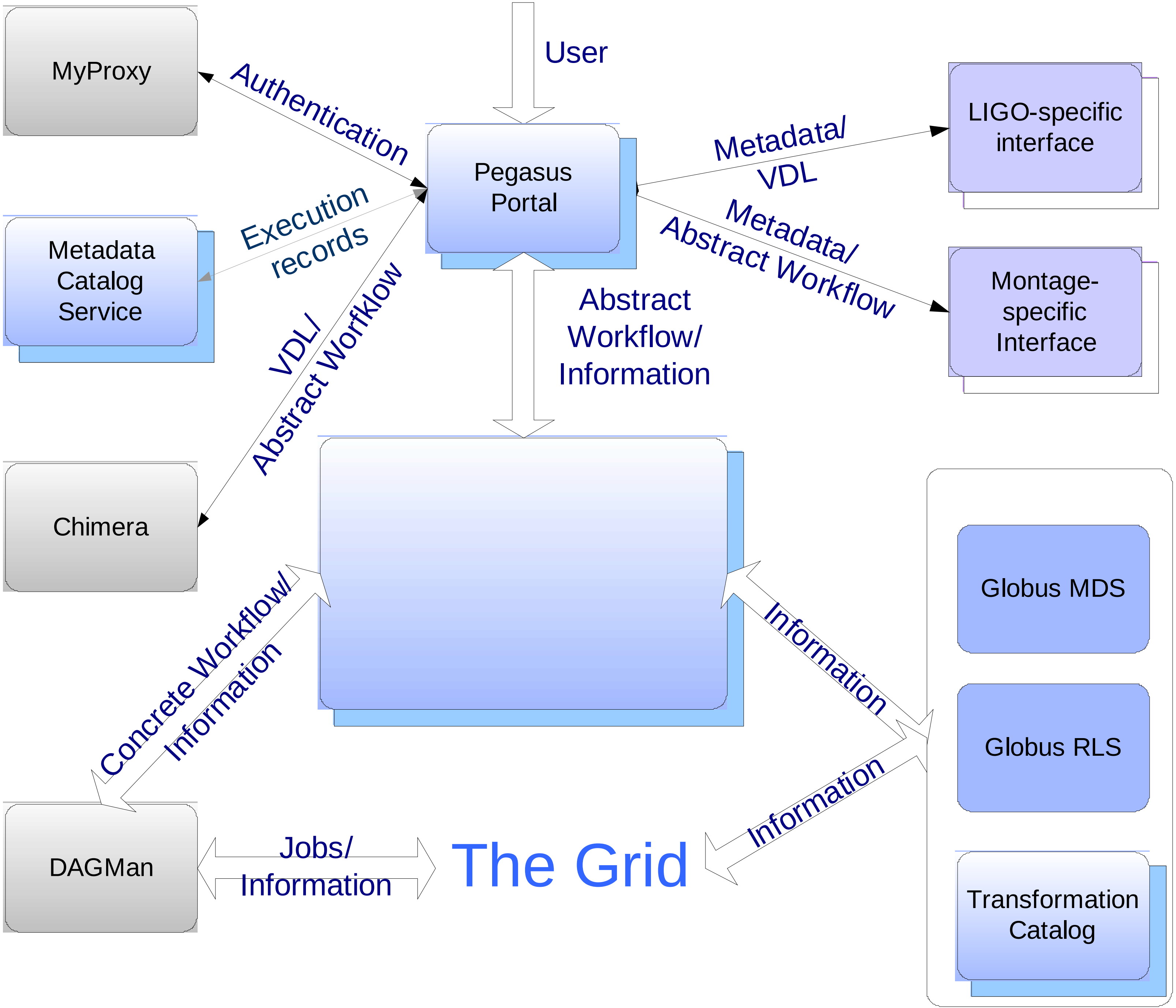
Montage Acknowledgments

- Bruce Berriman, John Good, Anastasia Laity, Caltech/IPAC
- Joseph C. Jacob, Daniel S. Katz, JPL
- http://montage.ipac. caltech.edu/

- Testbed for Montage: Condor pools at USC/ISI, UW Madison, and Teragrid resources at NCSA, PSC, and SDSC.

Montage is funded by the National Aeronautics and Space Administration's Earth Science Technology Office, Computational Technologies Project, under Cooperative Agreement Number NCC5-626 between NASA and the California Institute of Technology.

Simplified View of SC 2003 Portal



MyProxy

Authentication

User

Pegasus Portal

Metadata/ VDL

LIGO-specific interface

Metadata Catalog Service

Execution records

Metadata/ Abstract Workflow

Montage-specific Interface

VDL/ Abstract Workflow

Abstract Workflow/ Information

Chimera

Concrete Workflow/ Information

Information

Globus MDS

Globus RLS

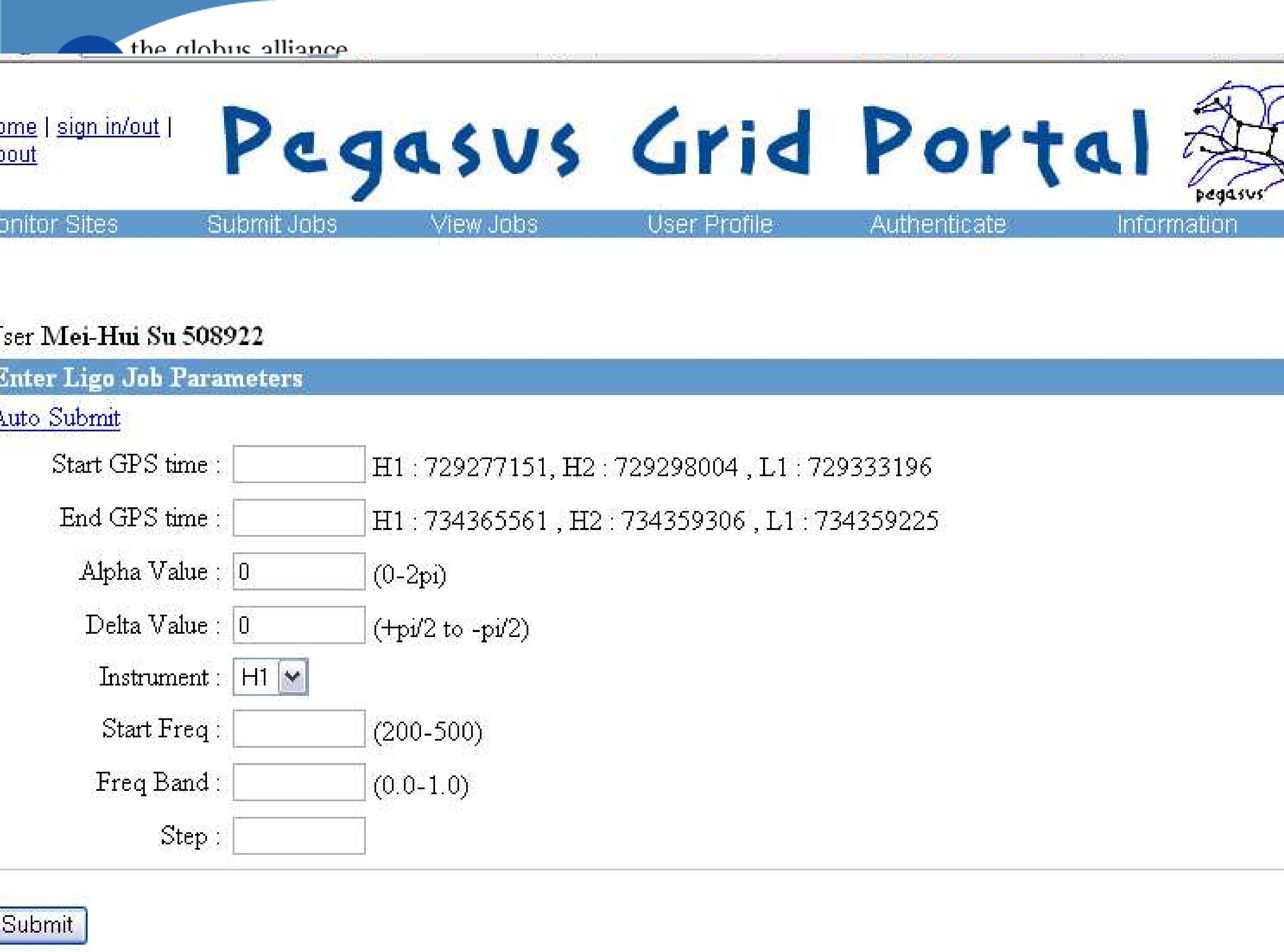
Information

DAGMan

Jobs/ Information

The Grid

Transformation Catalog



'ser **Mei-Hui Su 508922**

Enter Ligo Job Parameters

Auto Submit

Start GPS time : H1 : 729277151, H2 : 729298004 , L1 : 729333196

End GPS time : H1 : 734365561 , H2 : 734359306 , L1 : 734359225

Alpha Value : 0 (0-2pi)

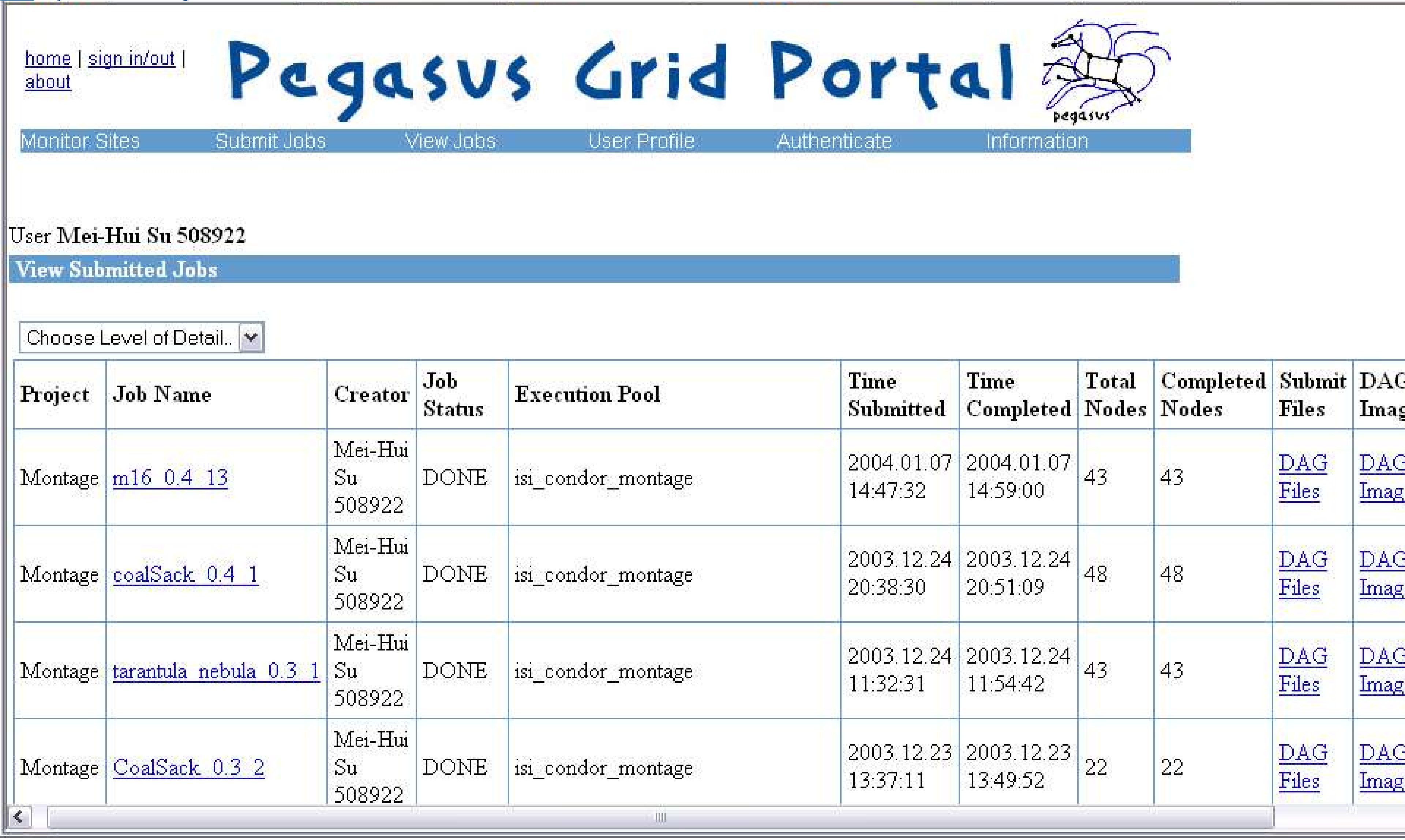
Delta Value : 0 (+pi/2 to -pi/2)

Instrument : H1

Start Freq : (200-500)

Freq Band : (0.0-1.0)

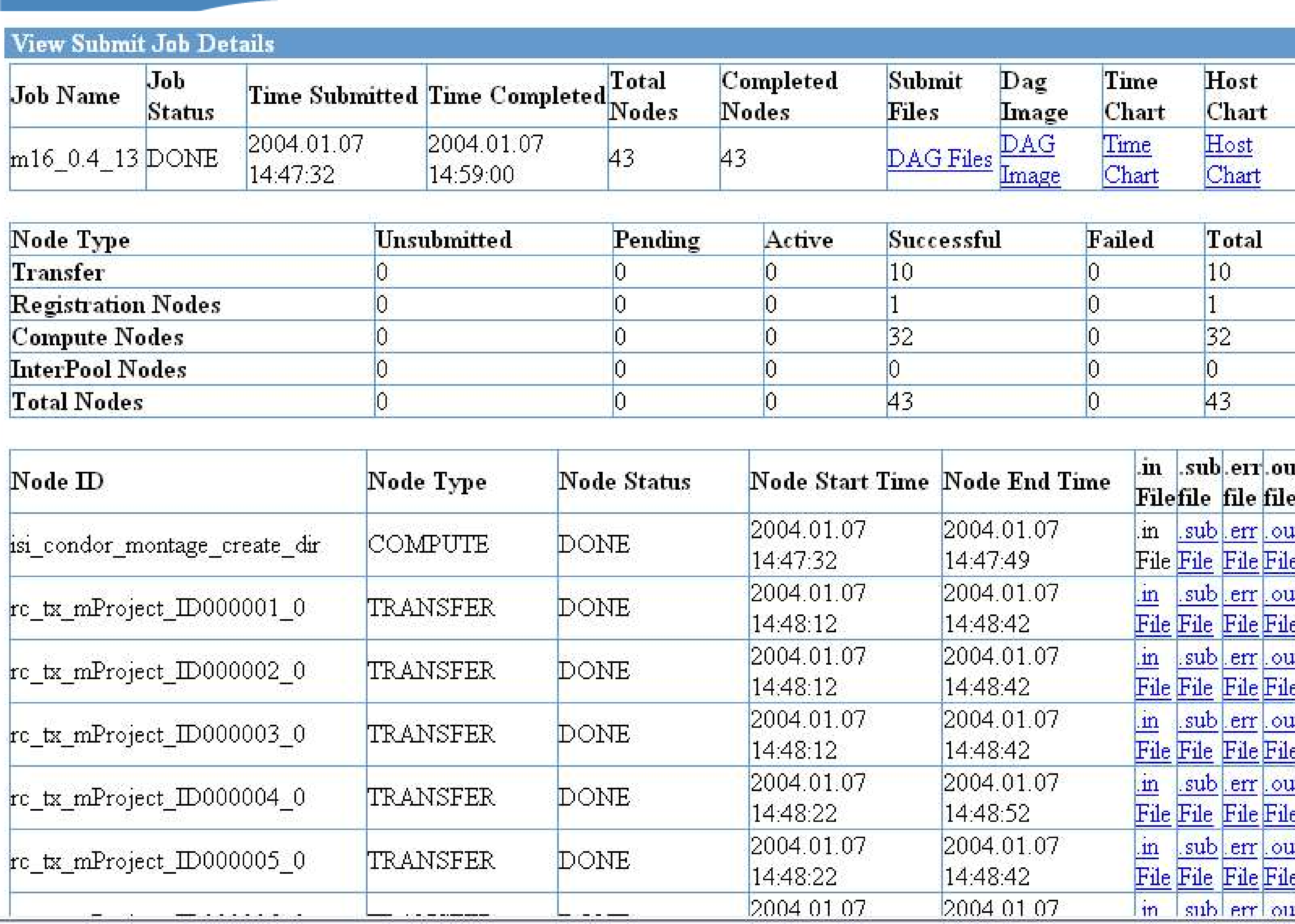
Step :



User **Mei-Hui Su 508922**

View Submitted Jobs

Project	Job Name	Creator	Job Status	Execution Pool	Time Submitted	Time Completed	Total Nodes	Completed Nodes	Submit Files	DAG Imag
Montage	m16_0.4_13	Mei-Hui Su 508922	DONE	isi_condor_montage	2004.01.07 14:47:32	2004.01.07 14:59:00	43	43	DAG Files	DAG Imag
Montage	coalSack_0.4_1	Mei-Hui Su 508922	DONE	isi_condor_montage	2003.12.24 20:38:30	2003.12.24 20:51:09	48	48	DAG Files	DAG Imag
Montage	tarantula_nebula_0.3_1	Mei-Hui Su 508922	DONE	isi_condor_montage	2003.12.24 11:32:31	2003.12.24 11:54:42	43	43	DAG Files	DAG Imag
Montage	CoalSack_0.3_2	Mei-Hui Su 508922	DONE	isi_condor_montage	2003.12.23 13:37:11	2003.12.23 13:49:52	22	22	DAG Files	DAG Imag

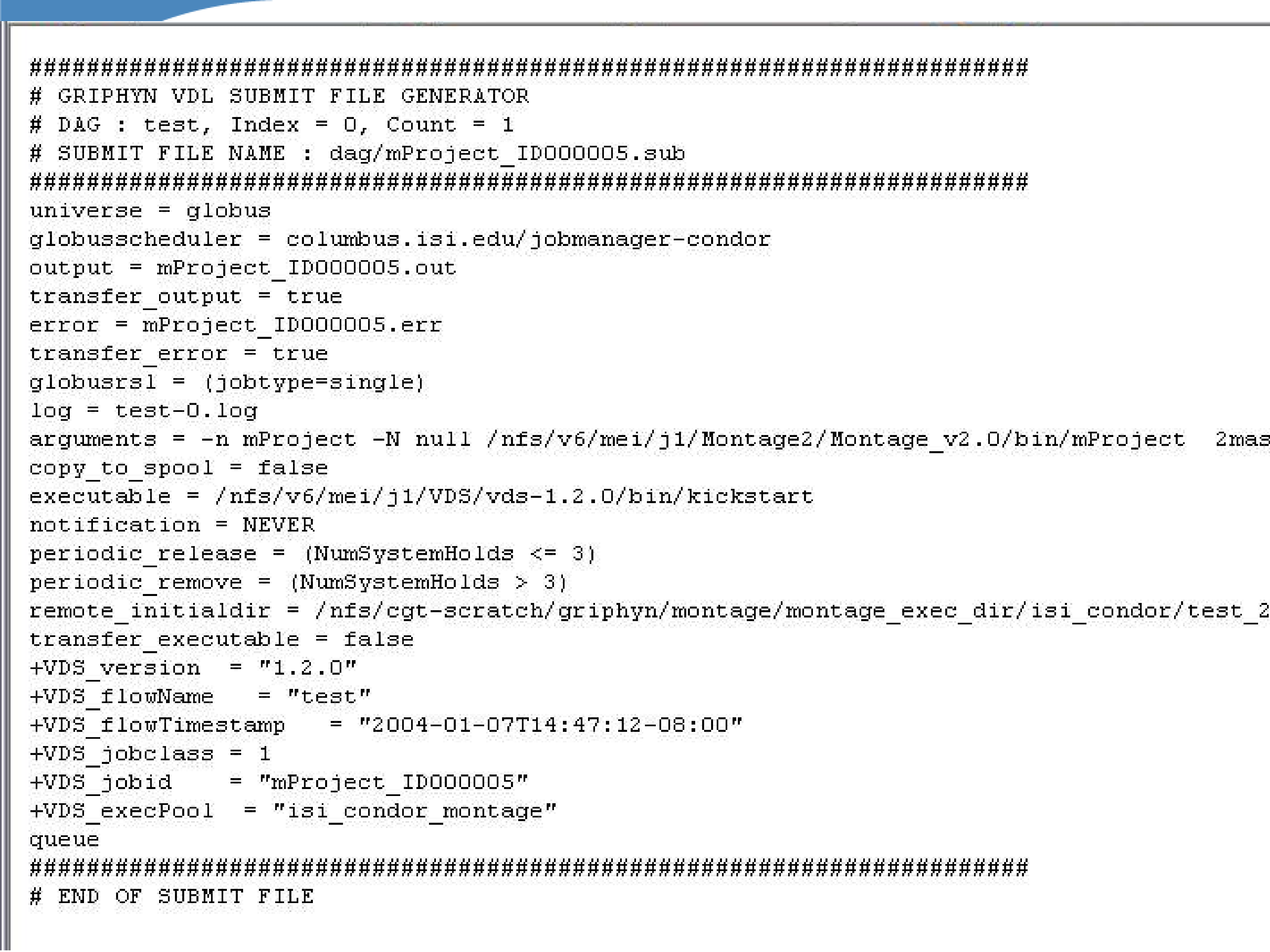


View Submit Job Details

Job Name	Job Status	Time Submitted	Time Completed	Total Nodes	Completed Nodes	Submit Files	Dag Image	Time Chart	Host Chart
m16_0.4_13	DONE	2004.01.07 14:47:32	2004.01.07 14:59:00	43	43	DAG Files	DAG Image	Time Chart	Host Chart

Node Type	Unsubmitted	Pending	Active	Successful	Failed	Total
Transfer	0	0	0	10	0	10
Registration Nodes	0	0	0	1	0	1
Compute Nodes	0	0	0	32	0	32
InterPool Nodes	0	0	0	0	0	0
Total Nodes	0	0	0	43	0	43

Node ID	Node Type	Node Status	Node Start Time	Node End Time	.in File	.sub file	.err file	.ou file
isi_condor_montage_create_dir	COMPUTE	DONE	2004.01.07 14:47:32	2004.01.07 14:47:49	.in File	.sub File	.err File	.ou File
rc_tx_mProject_ID000001_0	TRANSFER	DONE	2004.01.07 14:48:12	2004.01.07 14:48:42	.in File	.sub File	.err File	.ou File
rc_tx_mProject_ID000002_0	TRANSFER	DONE	2004.01.07 14:48:12	2004.01.07 14:48:42	.in File	.sub File	.err File	.ou File
rc_tx_mProject_ID000003_0	TRANSFER	DONE	2004.01.07 14:48:12	2004.01.07 14:48:42	.in File	.sub File	.err File	.ou File
rc_tx_mProject_ID000004_0	TRANSFER	DONE	2004.01.07 14:48:22	2004.01.07 14:48:52	.in File	.sub File	.err File	.ou File
rc_tx_mProject_ID000005_0	TRANSFER	DONE	2004.01.07 14:48:22	2004.01.07 14:48:42	.in File	.sub File	.err File	.ou File
			2004.01.07	2004.01.07	.in	.sub	.err	.ou



```
######################################################################
# GRIPHYN VDL SUBMIT FILE GENERATOR
# DAG : test, Index = 0, Count = 1
# SUBMIT FILE NAME : dag/mProject_ID000005.sub
######################################################################
universe = globus
globusscheduler = columbus.isi.edu/jobmanager-condor
output = mProject_ID000005.out
transfer_output = true
error = mProject_ID000005.err
transfer_error = true
globusrsl = (jobtype=single)
log = test-0.log
arguments = -n mProject -N null /nfs/v6/mei/j1/Montage2/Montage_v2.0/bin/mProject  2mas
copy_to_spool = false
executable = /nfs/v6/mei/j1/VDS/vds-1.2.0/bin/kickstart
notification = NEVER
periodic_release = (NumSystemHolds <= 3)
periodic_remove = (NumSystemHolds > 3)
remote_initialdir = /nfs/cgt-scratch/griphyn/montage/montage_exec_dir/isi_condor/test_2
transfer_executable = false
+VDS_version  = "1.2.0"
+VDS_flowName   = "test"
+VDS_flowTimestamp   = "2004-01-07T14:47:12-08:00"
+VDS_jobclass = 1
+VDS_jobid    = "mProject_ID000005"
+VDS_execPool  = "isi_condor_montage"
queue
######################################################################
# END OF SUBMIT FILE
```

Conclusions

- Pegasus maps complex workflows onto the Grid
- Uses Grid information services to find resources, data and executables
- Reduces the workflow based on existing intermediate products
- Used in many applications
- Part of GriPhyN's Virtual Data Toolkit

Future Directions

- Incorporate AI-planning technologies in production software (Virtual Data Toolkit)
- Investigate various scheduling techniques
- Investigating fault tolerance issues
  - Selecting resources based on their reliability
  - Responding to failures
- http://pegasus.isi.edu