

# Data Placement for Scientific Applications in Distributed Environments

**Ann Chervenak<sup>1</sup>, Ewa Deelman<sup>1</sup>, Miron Livny<sup>2</sup>, Mei-Hui Su<sup>1</sup>,  
Rob Schuler<sup>1</sup>, Shishir Bharathi<sup>1</sup>, Gaurang Mehta<sup>1</sup>, Karan Vahi<sup>1</sup>**

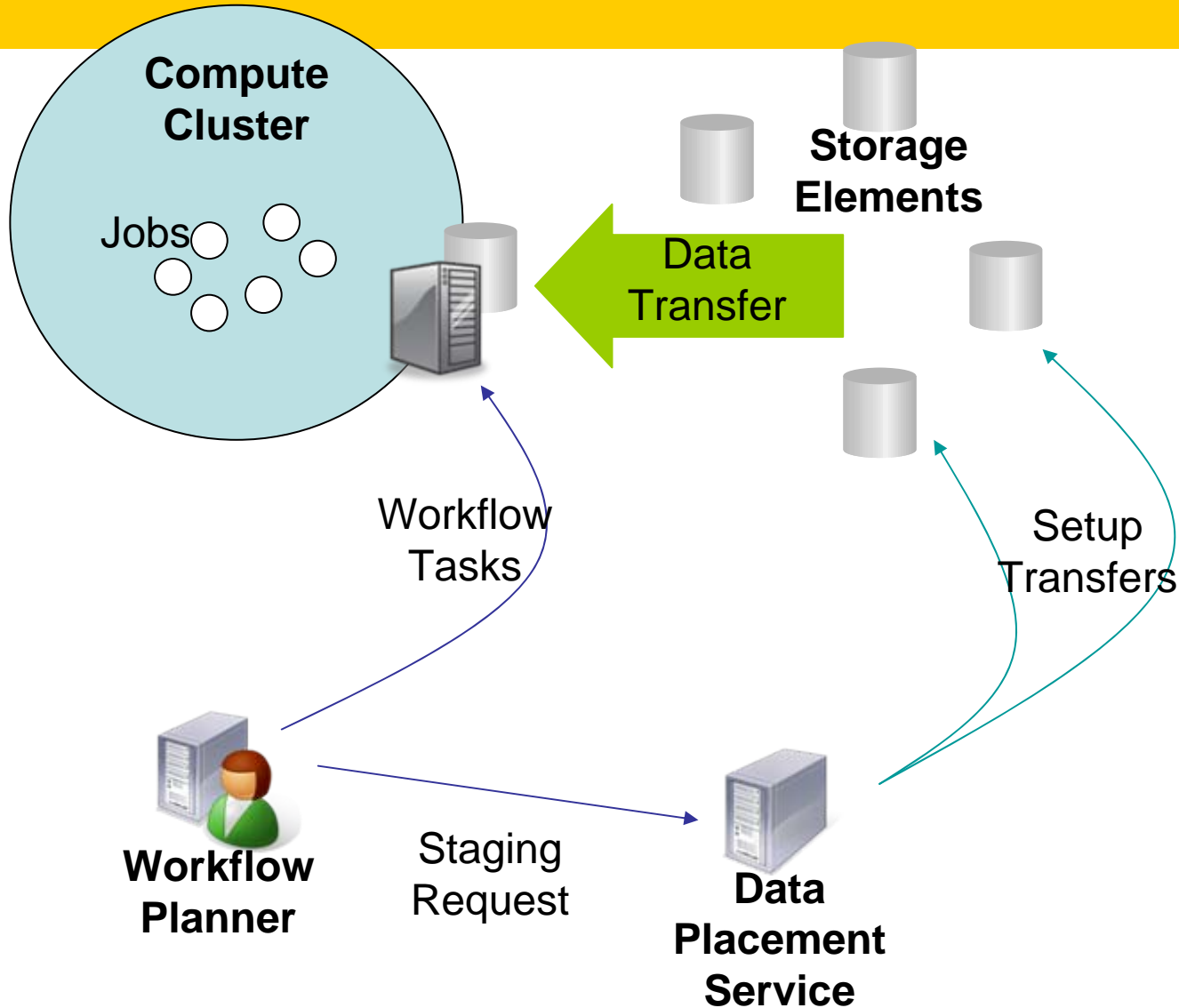
<sup>1</sup> USC Information Sciences Institute

<sup>2</sup> University of Wisconsin, Madison

# Motivation

- Scientific applications often perform complex computational analyses that consume and produce large data sets
- The placement of data onto storage systems can have a significant impact on
  - performance of applications
  - reliability and availability of data sets
- We want to identify data placement policies that distribute data sets so that they can be
  - staged into or out of computations efficiently
  - replicated to improve performance and reliability
- We study the relationship between **asynchronous data placement services** and **workflow management systems**
- Experimental evaluation demonstrates that good placement has the potential to significantly improve workflow execution performance

# Interaction Between Workflow Planner and Data Placement Service for Staging Data



- Example data placement service for high energy physics (PheDEx)
- Asynchronous data placement for scientific applications
- Approach: Integrate two components
  - Pegasus Workflow Management System
  - Data Replication Service
- Experimental evaluation: Workflow performance using asynchronous data placement
- Ongoing and Future Work



# Example Data Placement Service: Physics Experiment Data Export (PheDEX)

- Manages data distribution for CMS High Energy Physics Project
- High energy physics community has a hierarchical or tiered model for data distribution
  - Tier 0 at CERN: data collected, pre-processed, archived
  - Tier 1 sites: store & archive large subsets of Tier 0 data
  - Tier 2 sites: less storage, store a smaller subset of data
- Goal of PheDEX: automate data distribution processes
  - Staging data from tape to disk, wide area transfers, validation of transfers, migration from disk buffers to archive
- PheDEX system design involves agents running at each site, communicating through a central database
- PheDex supports:
  - initial "push-based" hierarchical distribution from Tier 0 site
  - subscription-based transfer of data
  - on-demand access to data by individual sites or scientists

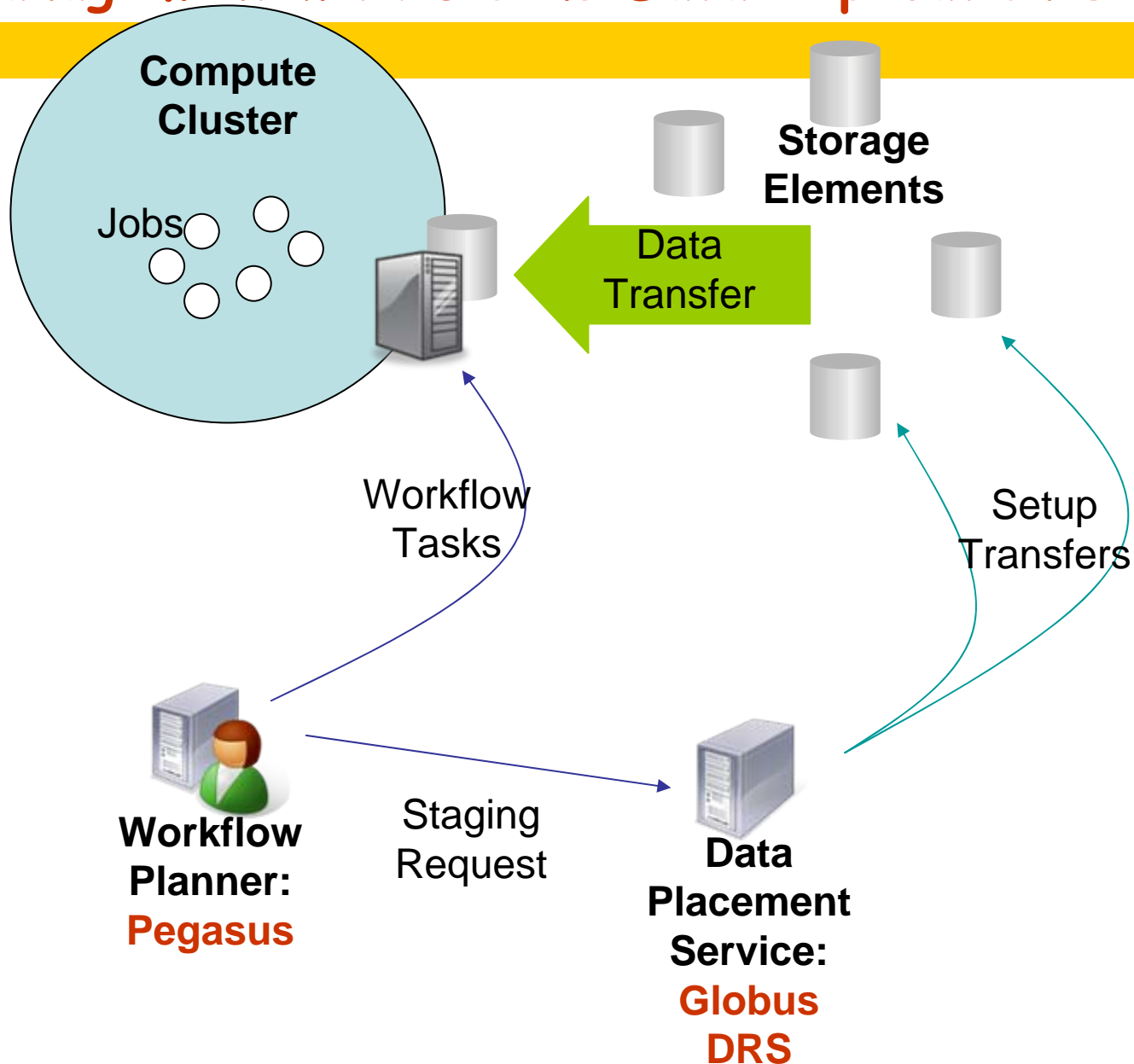
# Existing Data Placement Services

- PheDEX is one example of data management services for science
  - Others include Lightweight Data Replicator (LDR) for the Laser Interferometer Gravitational Wave Observatory (LIGO) project
- Provide **asynchronous** data movement of large scientific data sets
  - Terabytes of data
  - Millions of files
- Disseminate subsets of data to multiple sites some time after data sets are produced
  - Based on VO policies, metadata queries, subscriptions, explicit data requests
- Stage data sets onto resources where scientists plan to run analyses

# Asynchronous Data Placement and Workflows

- Goal: Separate to the extent possible the activities of data placement and workflow management services
  - Placement of data items largely *asynchronous* with respect to workflow execution
- Placement operations are performed
  - as data sets become available
  - according to the policies of the Virtual Organization
- Workflow system can provide hints to placement service r.e. grouping of files, expected order of access, dependencies, etc.
- In contrast to many existing workflow systems
  - Explicitly stage data onto computational nodes before execution
- Some explicit data staging may still be required
- Data placement has potential to
  - Significantly reduce need for on-demand data staging
  - Improve workflow execution time

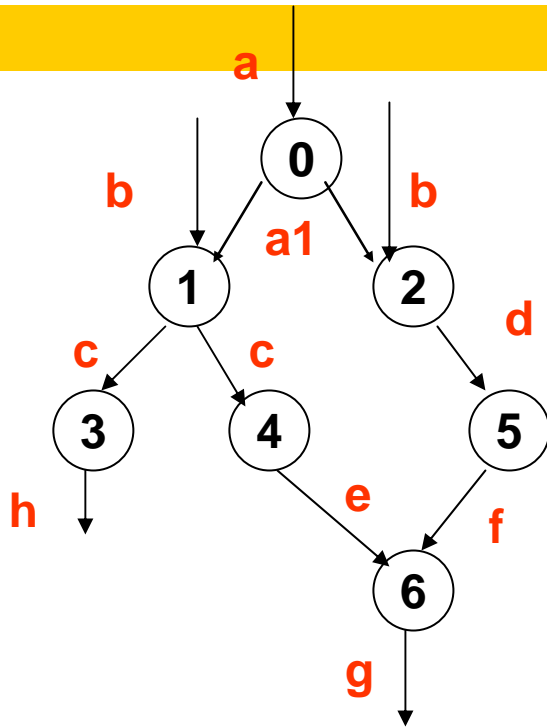
# Approach: Combine Pegasus Workflow Management with Globus Data Replication Service



# Pegasus Workflow Management System

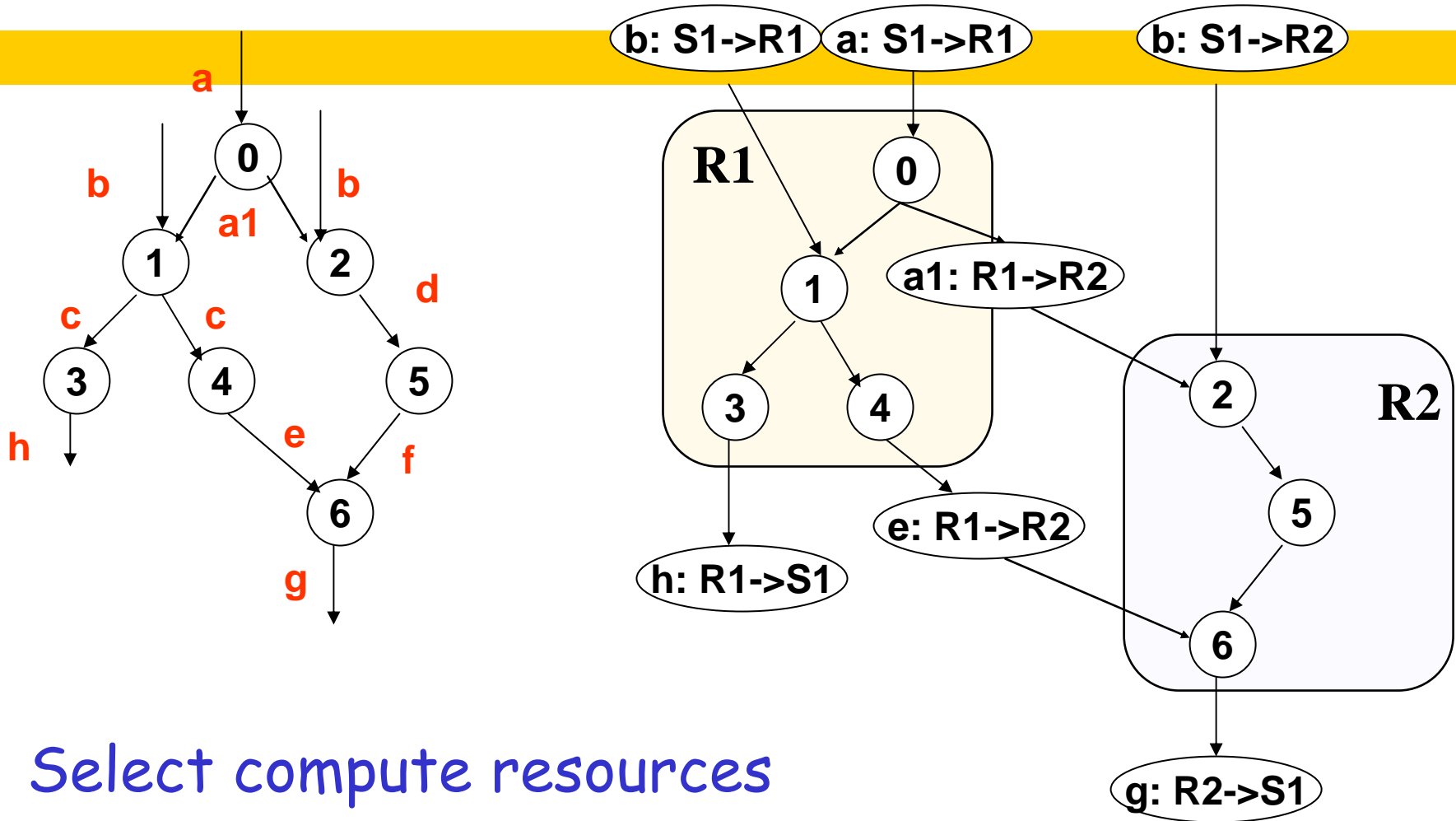
- Maps from high-level, resource-independent workflow descriptions to executable workflows
  - Finds appropriate resources
  - Finds data described in the workflow
  - Finds appropriate executables and stages them in if necessary
- Creates an executable workflow
  - Schedules the computations and creates computational nodes
  - Adds data transfer and data registration nodes
  - Performs optimizations
    - Node clustering
    - Dynamic data cleanup
- Relies on Condor DAGMan for correct, scalable, and reliable execution
- Used in astronomy, earthquake science, gravitational-wave science, and other disciplines

# Pegasus Mapping



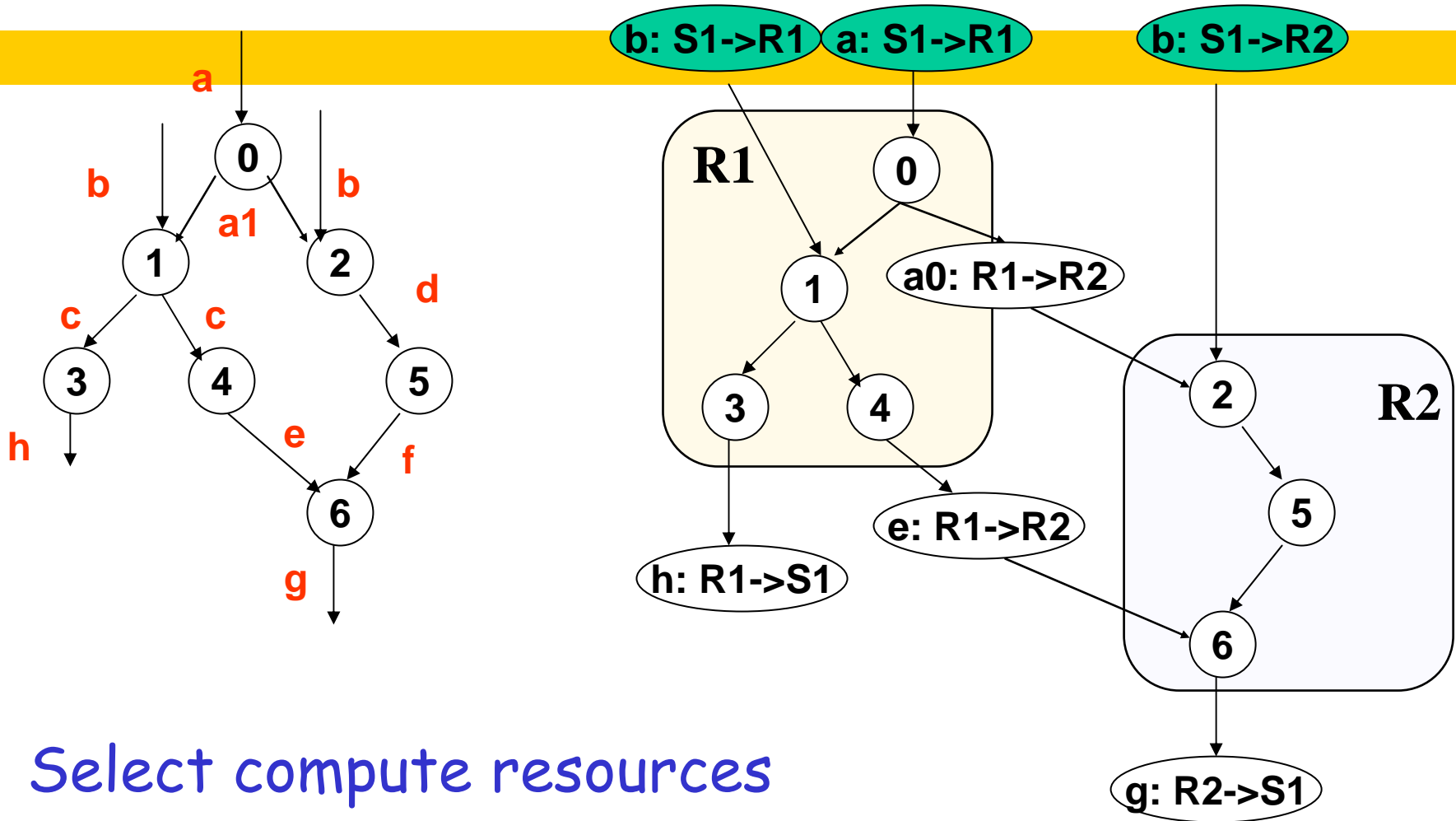
- Select compute resources
- Select data sources
- Add data stage-in and data stage-out nodes

# Pegasus Mapping



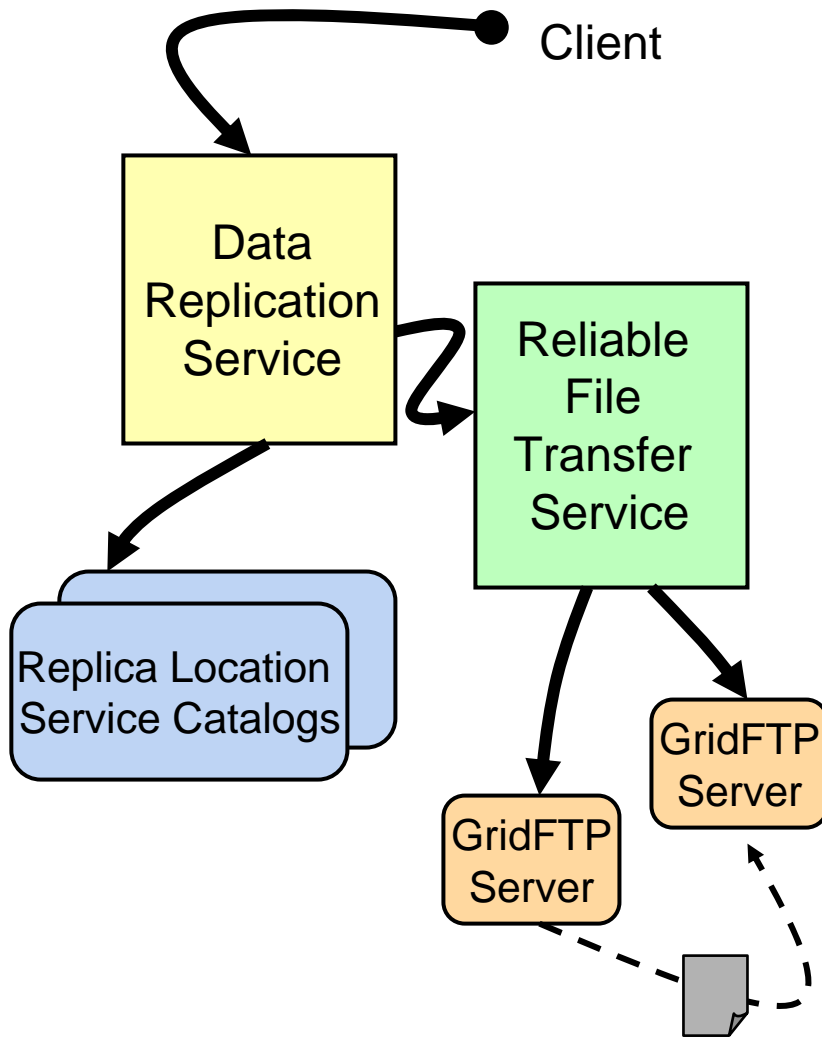
- Select compute resources
- Select data sources
- Add data stage-in and data stage-out nodes

# Pegasus Mapping



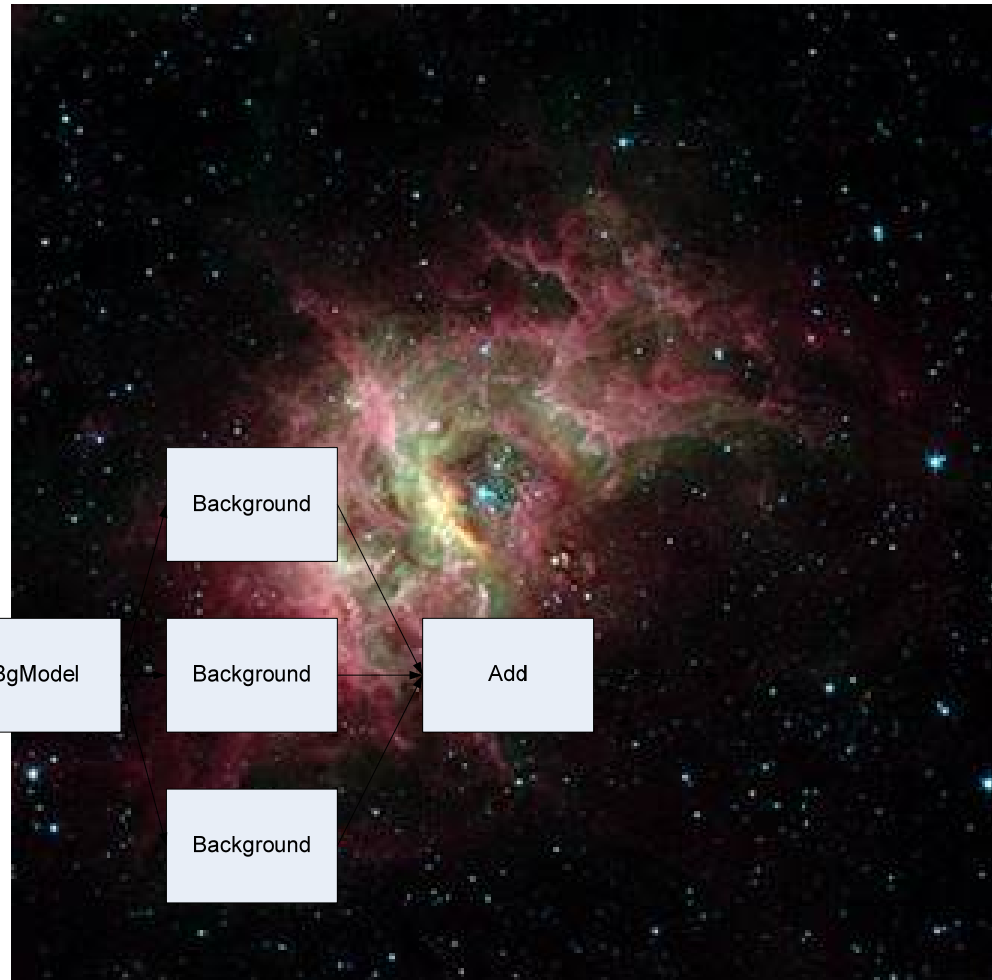
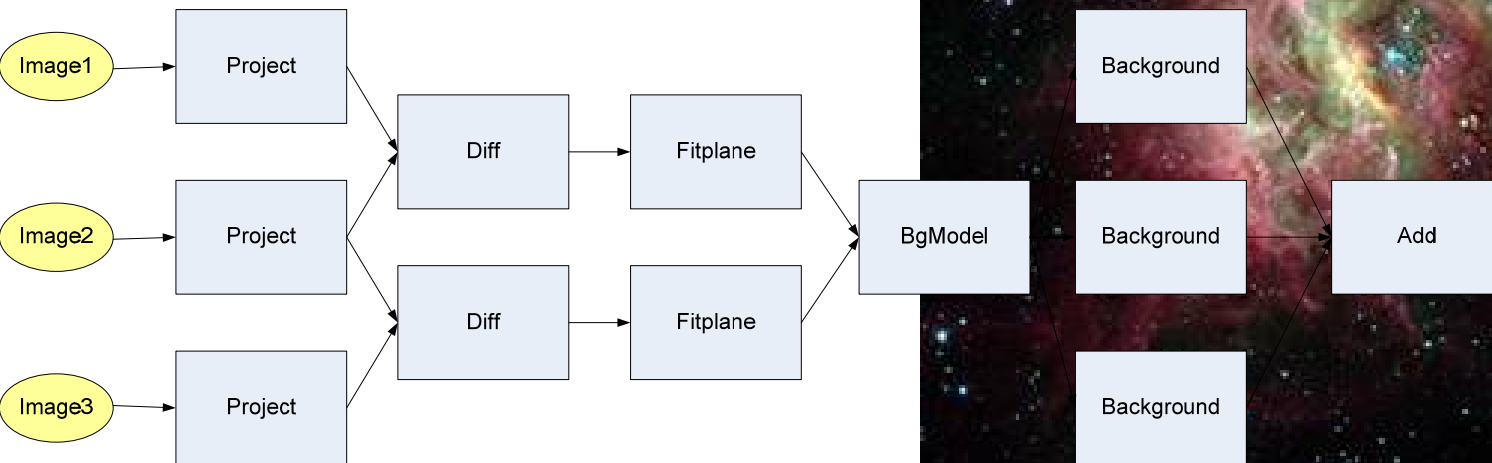
- Select compute resources
- Select data sources
- Add data stage-in and data stage-out nodes

# Data Replication Service



- Service that replicates a set of files at a site and registers them in catalog for later discovery
- Discovers replicas (possible source files) that exist in the Grid
  - Uses Globus Replica Location Service
- Select among source files
- Invoke Globus Reliable File Transfer Service to copy data
  - Uses GridFTP Data Transfer service
- Register new replicas in replica catalog

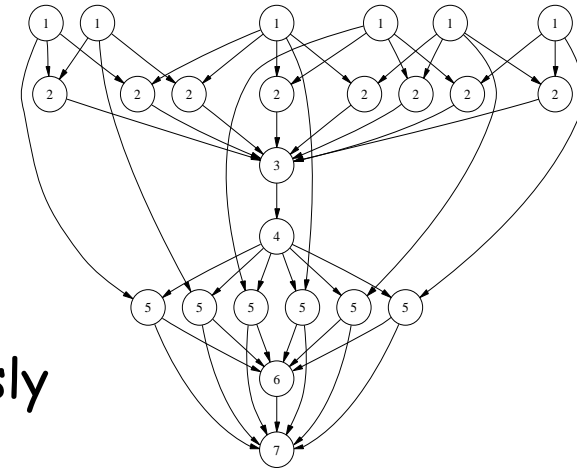
- Generates science-grade mosaics of the sky



Galactic Star Formation Region RCW 49

# Experimental Study

- Ran Montage workflows with different Mosaic degrees
  - Modified input data sizes to simulate larger data sets
- Pegasus workflows: compared explicit data staging tasks with workflows that check whether data has been staged asynchronously
- Use Globus DRS to stage input data asynchronously before workflow execution begins
- Workflows ran on a cluster with up to 50 available compute nodes, where each node has:
  - Dual Pentium III 1GHz processor
  - 1GByte of RAM
  - Debian Sarge 3.1 operating system



# Experimental Study (cont.)

- The data sets are staged onto a storage system associated with the cluster from a GridFTP server on the local area network
- We compare the time taken to:
  1. Stage data using the Data Replication Service
  2. Run the workflow when the data are already prestaged by DRS (requiring no additional data movement by Pegasus)
  3. Run the workflow when Pegasus manages the data staging explicitly
  4. Sum of DRS data staging time and the Pegasus execution time for prestaged data (corresponds to sequential invocation of these two services)

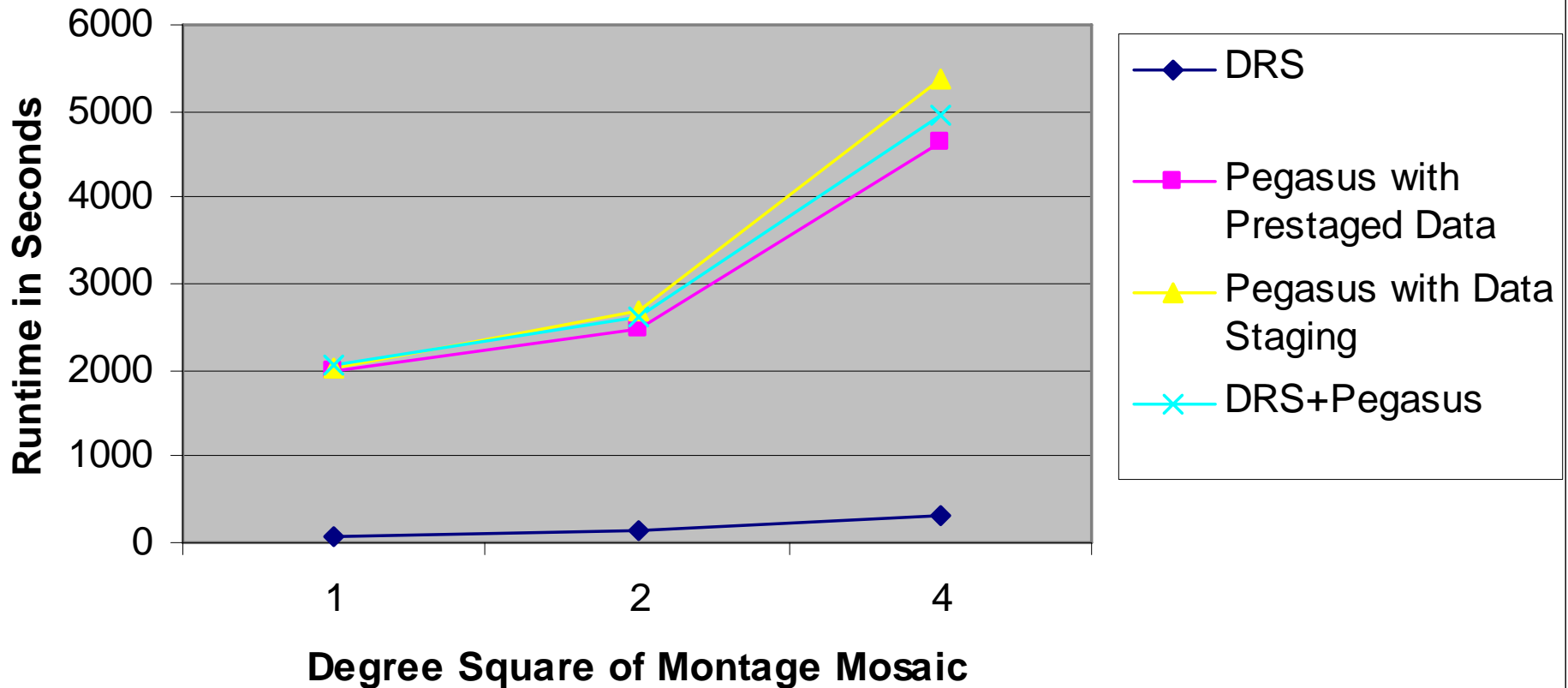
# Input Sizes for Experiments

Degree Square of Montage Mosaic	Number of input files for workflow execution		
	Default input size	With additional 2MB files	With additional 20MB files
1	50	95	95
2	166	318	318
4	648	1258	1258

Degree Square of Montage Mosaic	Total input size for workflow execution		
	Default	With additional 2MB files	With additional 20MB files
1	91 MBytes	182 MBytes	993 MBytes
2	307 MBytes	612 Mbytes	3.31 GBytes
4	1.2 GBytes	2.4 GBytes	13.2 GBytes

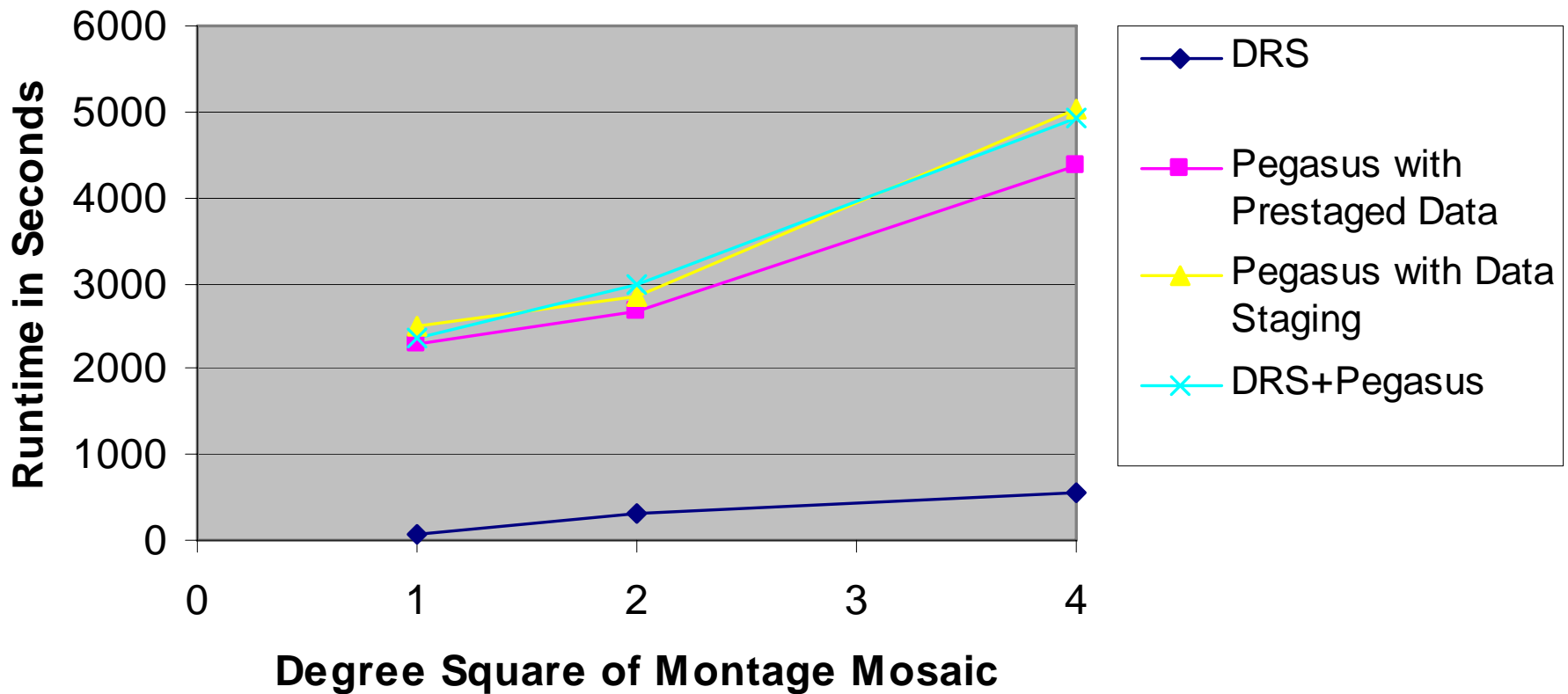
# Execution Times with Default Input Sizes

## Default Montage Workflow



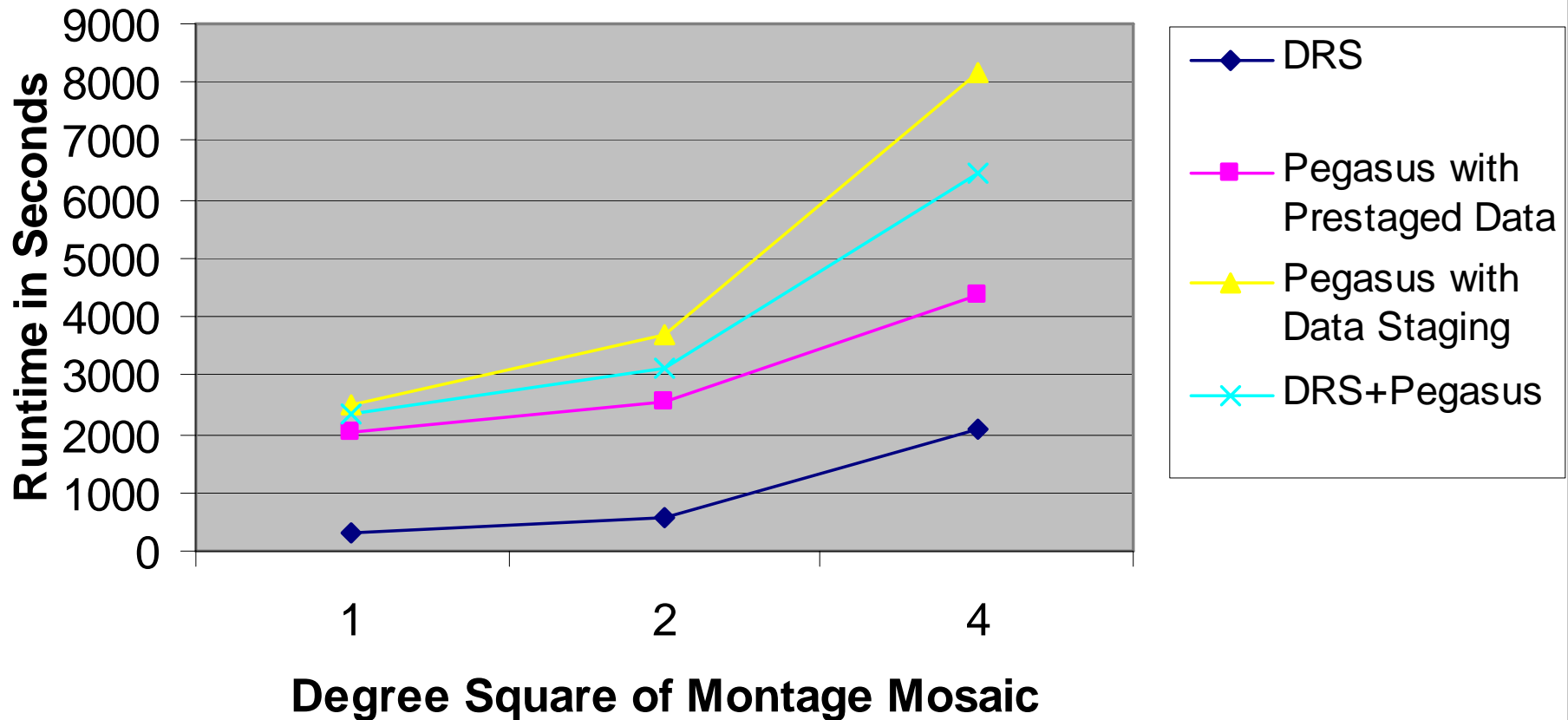
# Execution Times with Additional 2 MByte Input Files

## Montage Workflow with 2MB Additional Input Files



# Execution Times with Additional 20 MByte Input Files

## Montage Workflow with 20MB Additional Input Files



# Experiment Summary

- Largest workflow
  - Total input data size of 13.2 GBytes for a mosaic of 4 degree square
- Combination of prestaging data with DRS followed by workflow execution using Pegasus
  - Improves execution time approximately 21.4% over Pegasus performing explicit data staging
  - Avoid Condor queuing overheads for data staging tasks
- When data sets are completely prestaged before workflow execution begins
  - Execution time is reduced by over 46%
- Shows potential advantage of combining asynchronous data placement services with workflow management



# Ongoing and Future Work

First step towards understanding interplay between community-wide data placement services and community workflow management systems

Continued work on efficient stage-in and stage-out of data sets for workflows

- Initial data placement service, release October 2007
  - Will allow workflow manager or other client to specify movement of data sets, specify groups of files, change priorities for staging
- Simulation studies (based on GridSim) of workflow manager and placement service
  - Allow experimentation with different placement algorithms

Future work

- Placement services that replicate data for performance and reliability

# Acknowledgements

This work is supported by:

- The National Science Foundation under the grants CNS 0615412 and OCI 0534027
- The Department of Energy's Scientific Discovery through Advanced Computing II program under grant DE-FC02-06ER25757

Thanks to the Montage team (Bruce Berriman, John Good, and Daniel Katz) for their helpful discussions and the use the of the Montage codes and 2MASS data