Challenges of Managing Large-Scale Scientific Workflows In Distributed Environments

Ewa Deelman

Information Sciences Institute

University of Southern California

Scientific Applications Today

- Complex
  - Involve many computational steps
  - Require many (possibly diverse resources)
- Composed of individual application components
  - Components written by different individuals
  - Components require and generate large amounts of data
  - Components written in different languages
- Reuse of individual intermediate data products
- Need to keep track of how the data was produced

Ewa Deelman
deelman@isi.edu

Execution environment

- Many resources are available
- Resources are heterogeneous and distributed in the WAN
- Access to resources is often remote
- Resources come and go because of failure or policy changes

- Data is replicated at more than one location

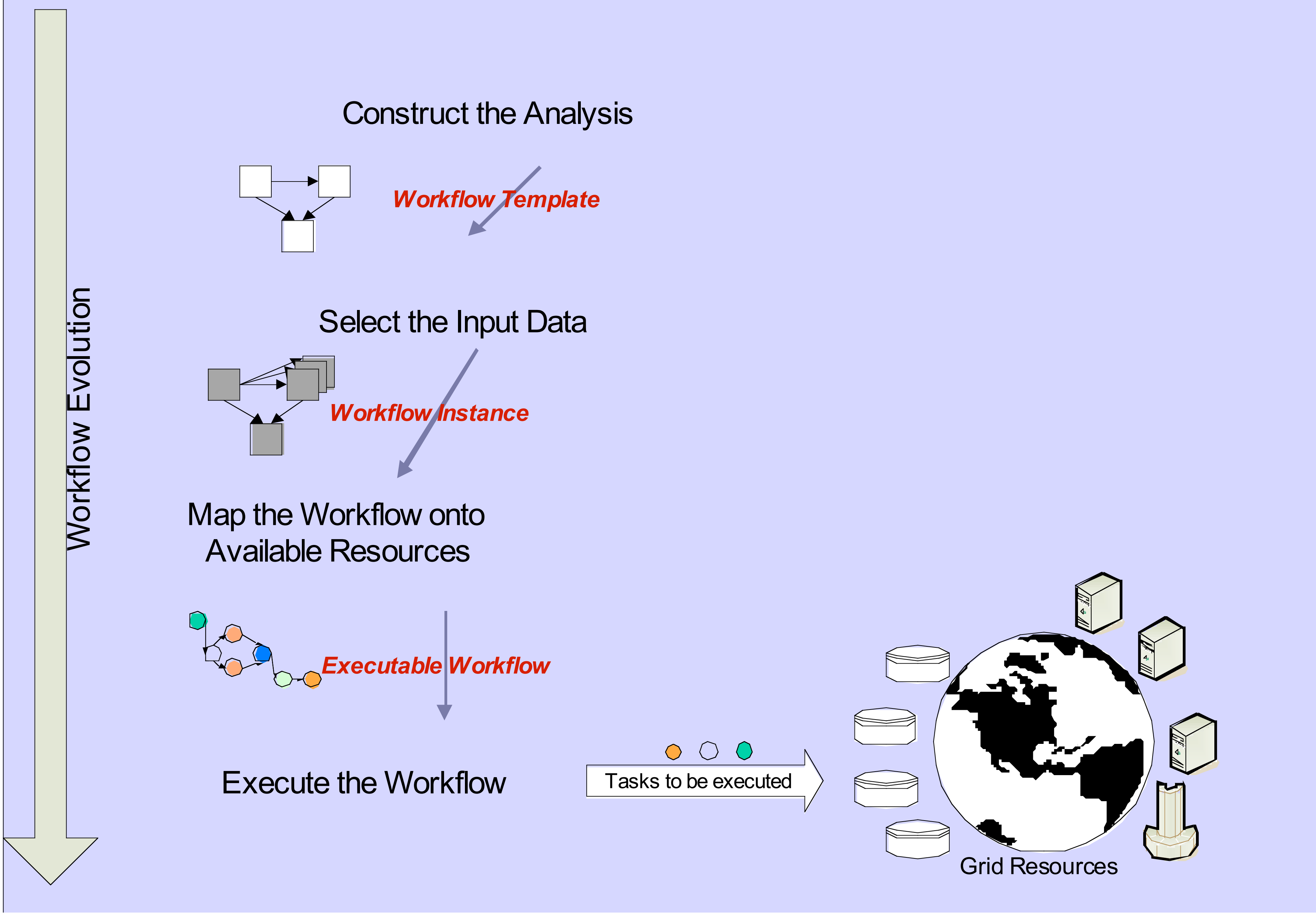
- Application components can be found at various locations or staged in on demand

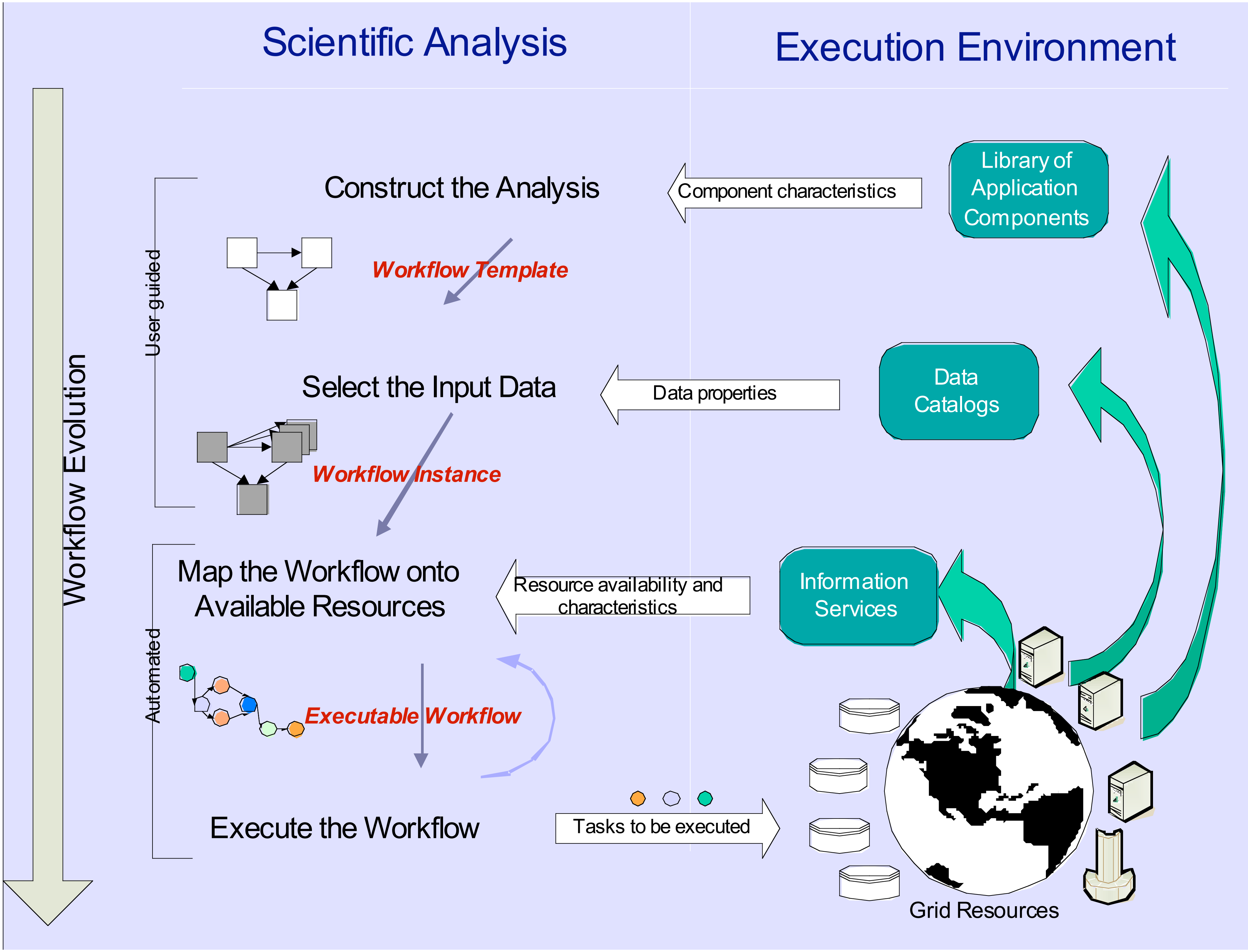
- **Problem:** How to compose and map applications onto the environment?
  - Efficiently &Reliably
- Structure the application as a workflow
  - Define the application components, the dependencies between them
- Tie the resources together into a Grid
- Develop a mapping strategy to map from the workflow description to the Grid resources

Outline

- Pegasus, mapping workflows onto the Grid
- Challenges in Workflow Performance
  - Workflow restructuring
  - Provisioning resources
  - Modeling and optimizing workflow component behavior
- Challenges in Workflow Reliability
  - Mapping portions of the workflow at a time
  - Efficient data handling
- Providing workflow mapping capabilities to a variety of workflow generation mechanism
- Application Experiences and Science Impacts
- Conclusions

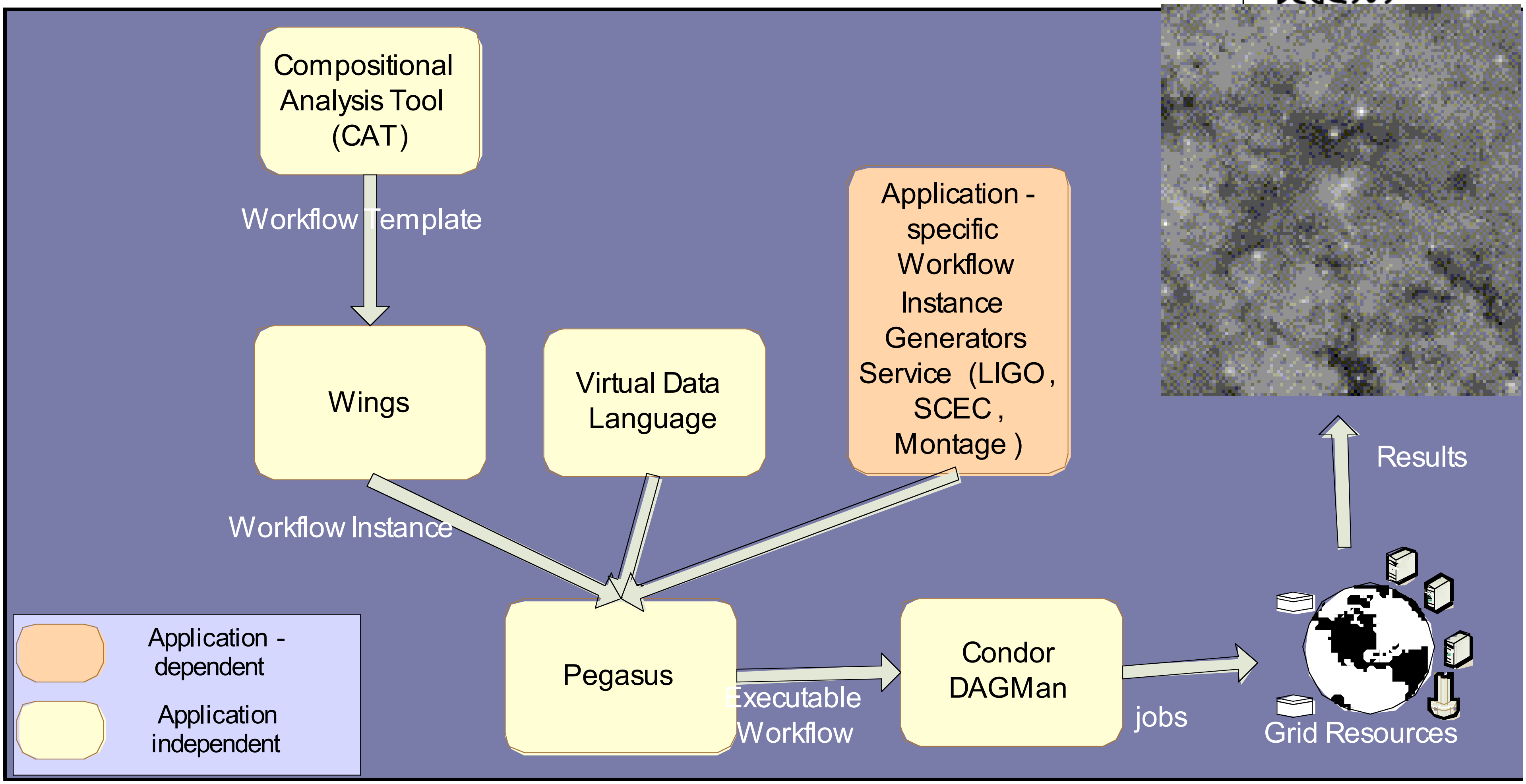
Scientific Analysis





Scientific Analysis
Execution Environment
Workflow Evolution
User guided
Automated
Construct the Analysis
Workflow Template
Select the Input Data
Workflow Instance
Map the Workflow onto Available Resources
Executable Workflow
Execute the Workflow
Component characteristics
Data properties
Resource availability and characteristics
Tasks to be executed
Library of Application Components
Data Catalogs
Information Services
Grid Resources

Workflow Instance Generation and Mapping



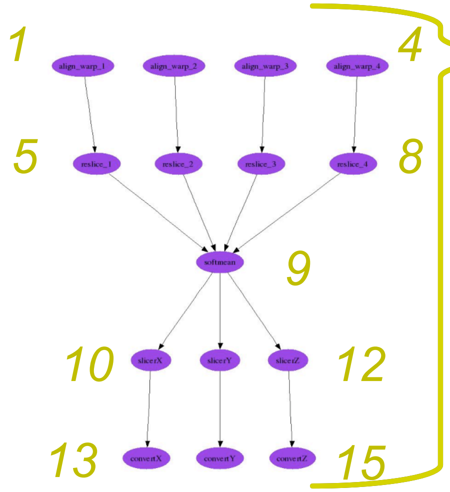
Pegasus: Planning for Execution in Grids

- Maps from a workflow instance to an executable workflow
- Automatically locates physical locations for both workflow components and data
- Finds appropriate resources to execute the components
- Reuses existing data products where applicable
- Publishes newly derived data products
  - Provides provenance information

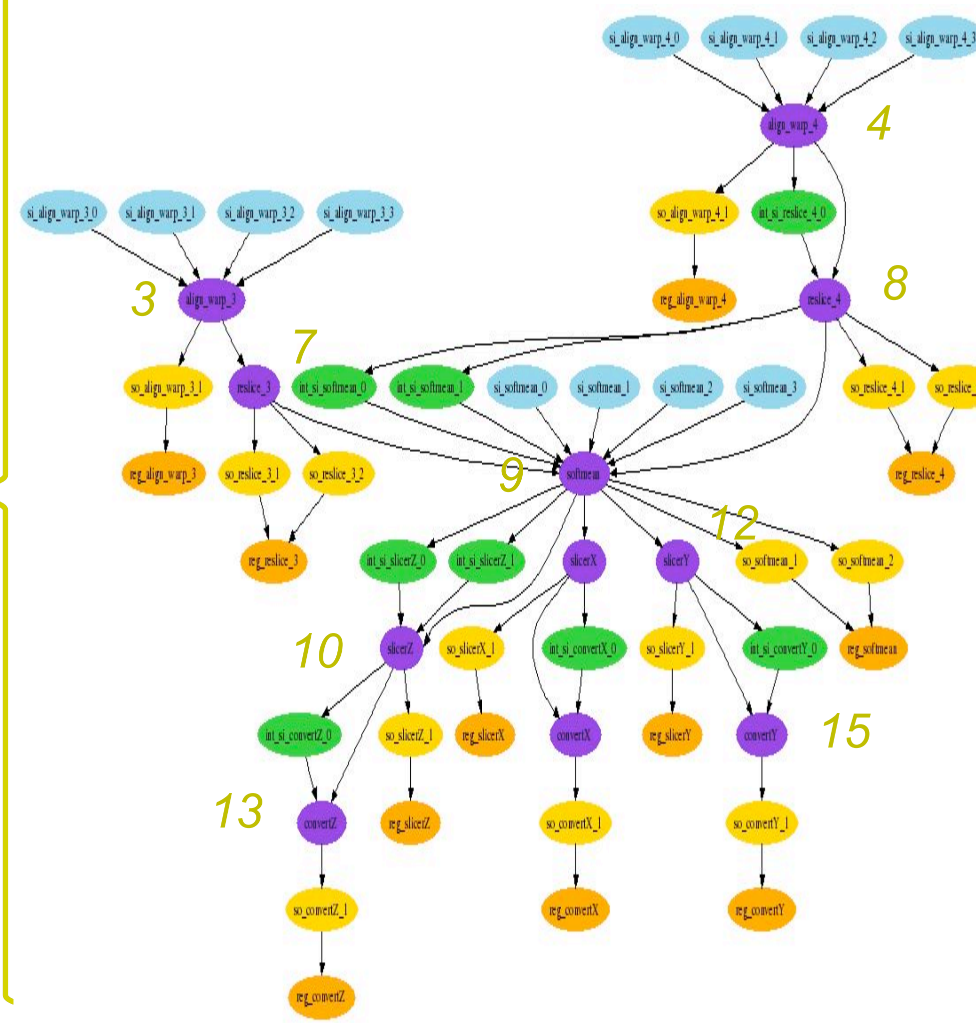
Information Components used by Pegasus

- Pegasus maintains interfaces to support a variety of information sources
- Information about resources
  - Globus Monitoring and Discovery Service (MDS)
    - Finds resource properties
    - Dynamic: load, queue length
    - Static: location of GridFTP server, RLS, etc
- Information about data location
  - Globus Replica Location Service
    - Locates data that may be replicated
    - Registers new data products
- Information about executables
  - Transformation Catalog

Pegasus Workflow Mapping



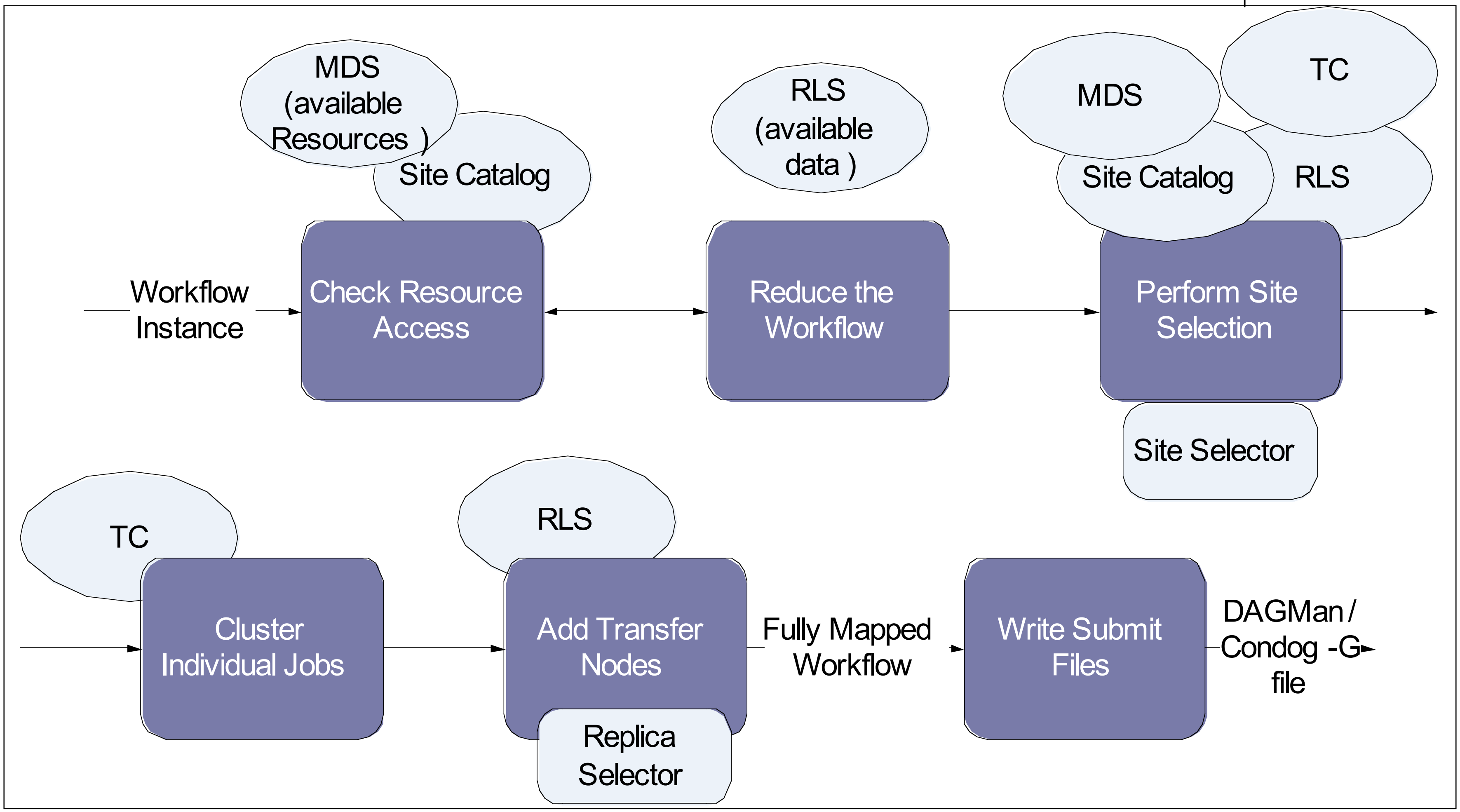
Original workflow: 15 compute nodes devoid of resource assignment



Resulting workflow mapped onto 3 Grid sites:

- 13 data stage-in nodes
- 11 compute nodes (4 reduced based on available intermediate data)
- 8 inter-site data transfers
- 14 data stage-out nodes to long-term storage
- 14 data registration nodes (data cataloging)

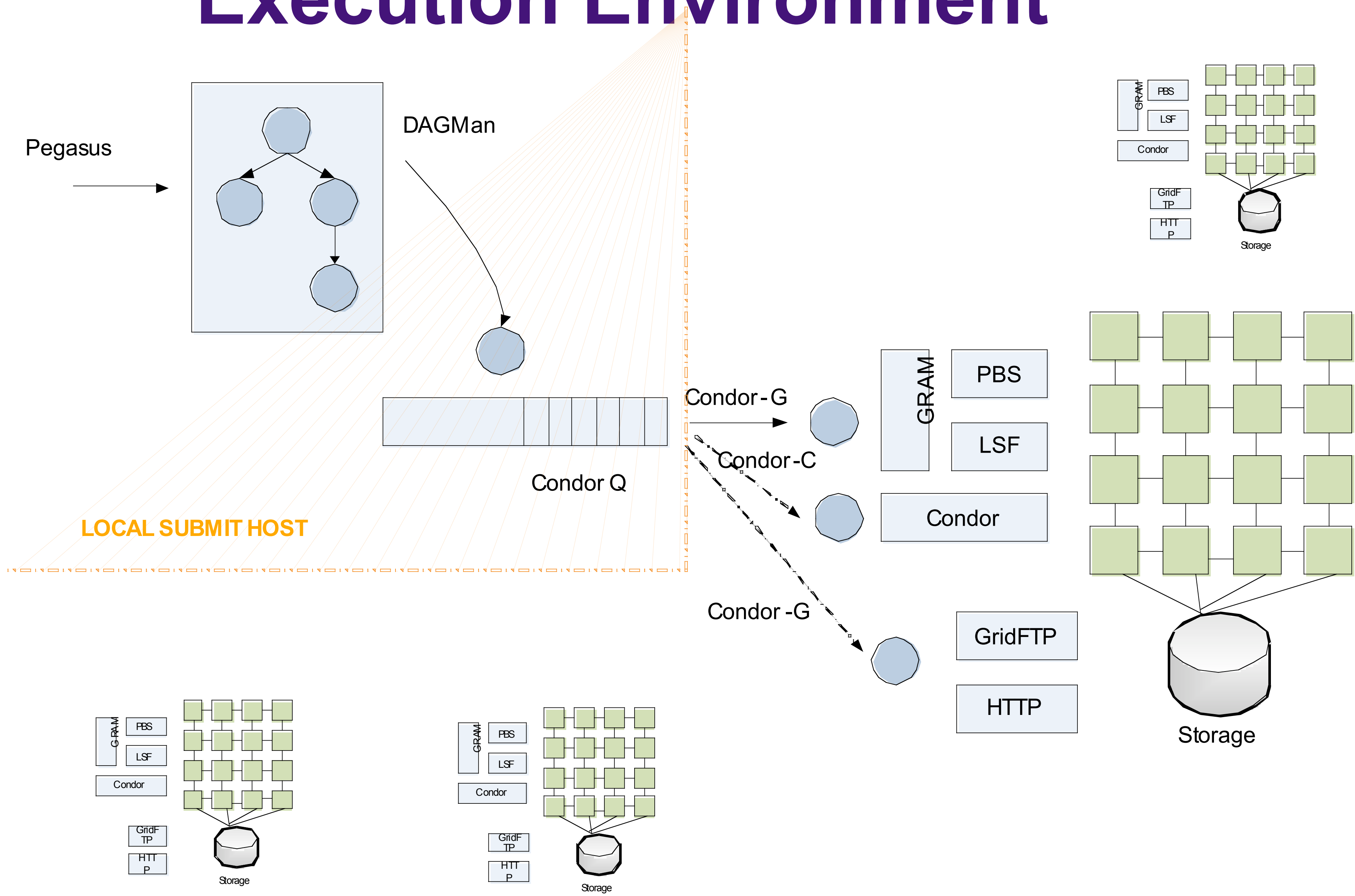
Pegasus Information Flow



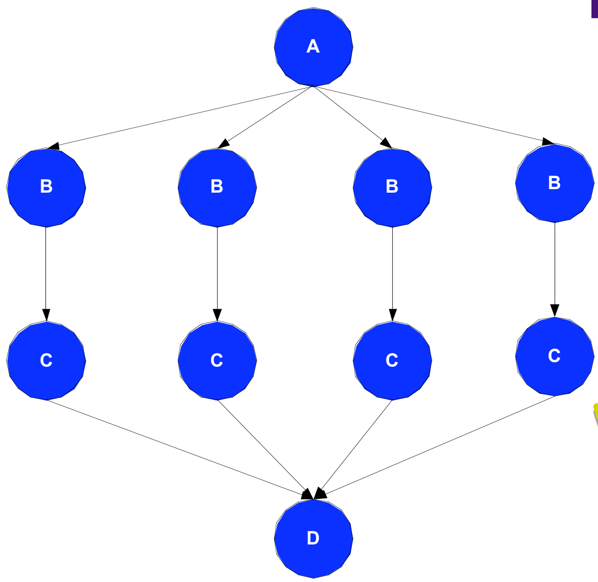
Outline

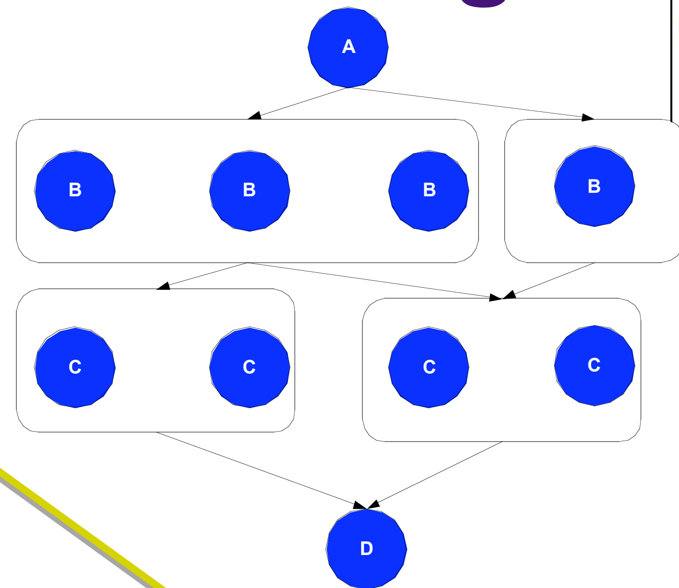
- Pegasus
- Challenges in Workflow Performance
  - Workflow restructuring
  - Provisioning resources
  - Modeling and optimizing workflow component behavior
- Challenges in Workflow Reliability
  - Mapping portions of the workflow at a time
  - Efficient data handling
- Providing workflow mapping capabilities to a variety of workflow generation mechanism
- Application Experiences and Science Impacts
- Conclusions

Execution Environment



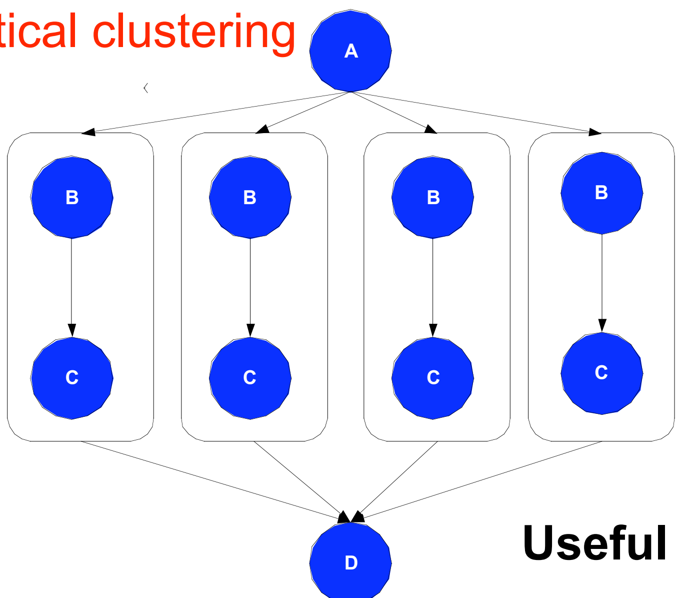
Node clustering



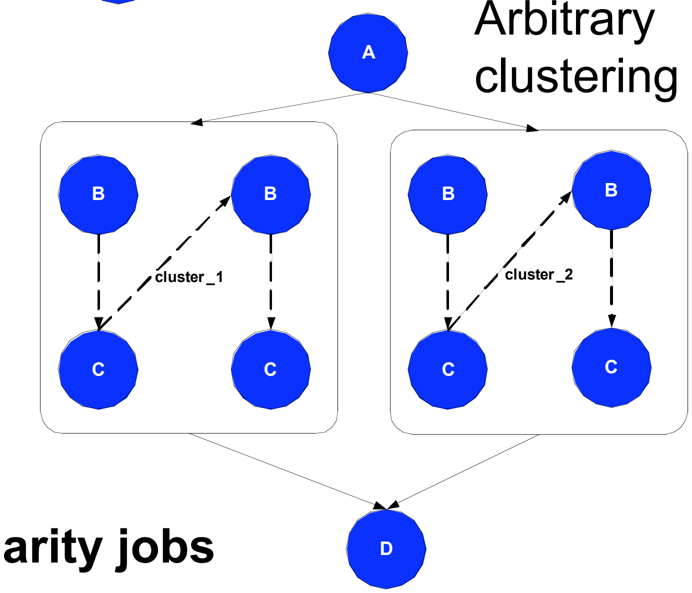


Level-based clustering

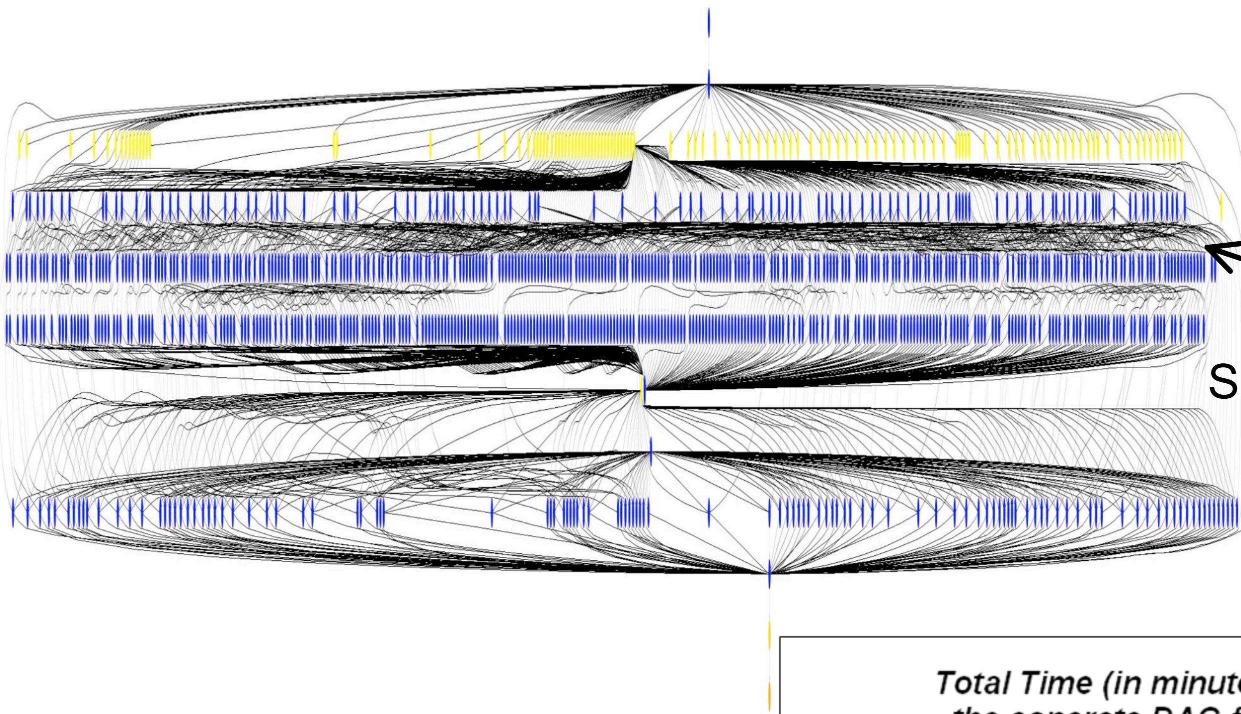
Vertical clustering



Arbitrary clustering

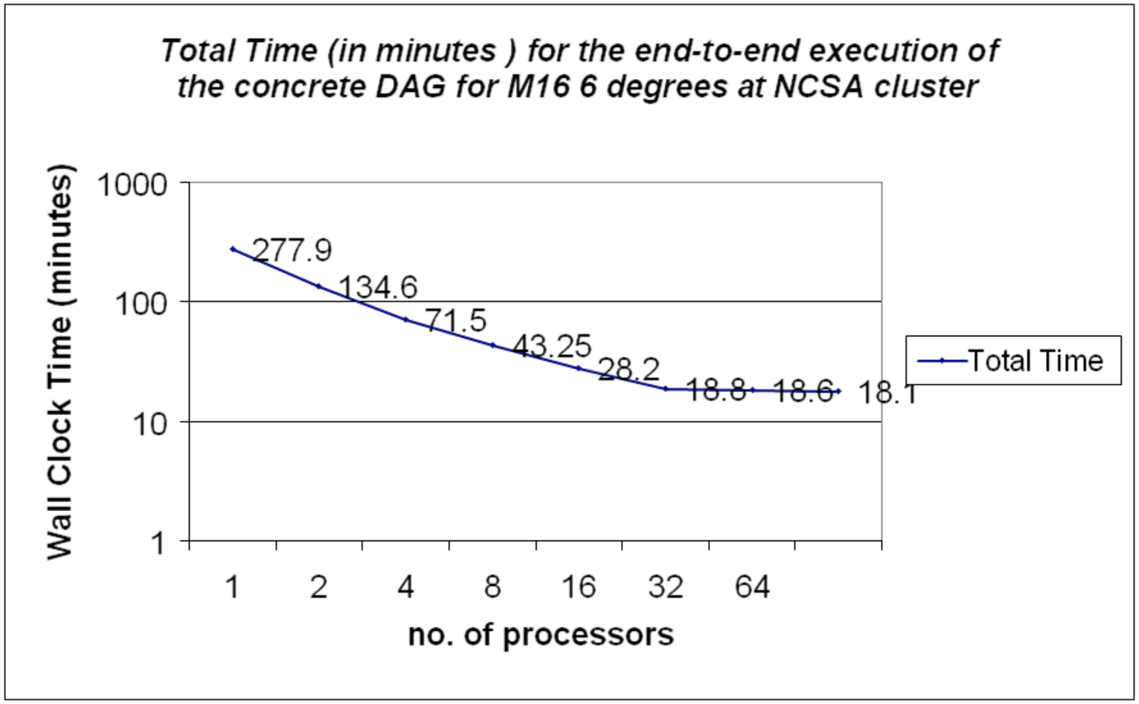


Useful for small granularity jobs

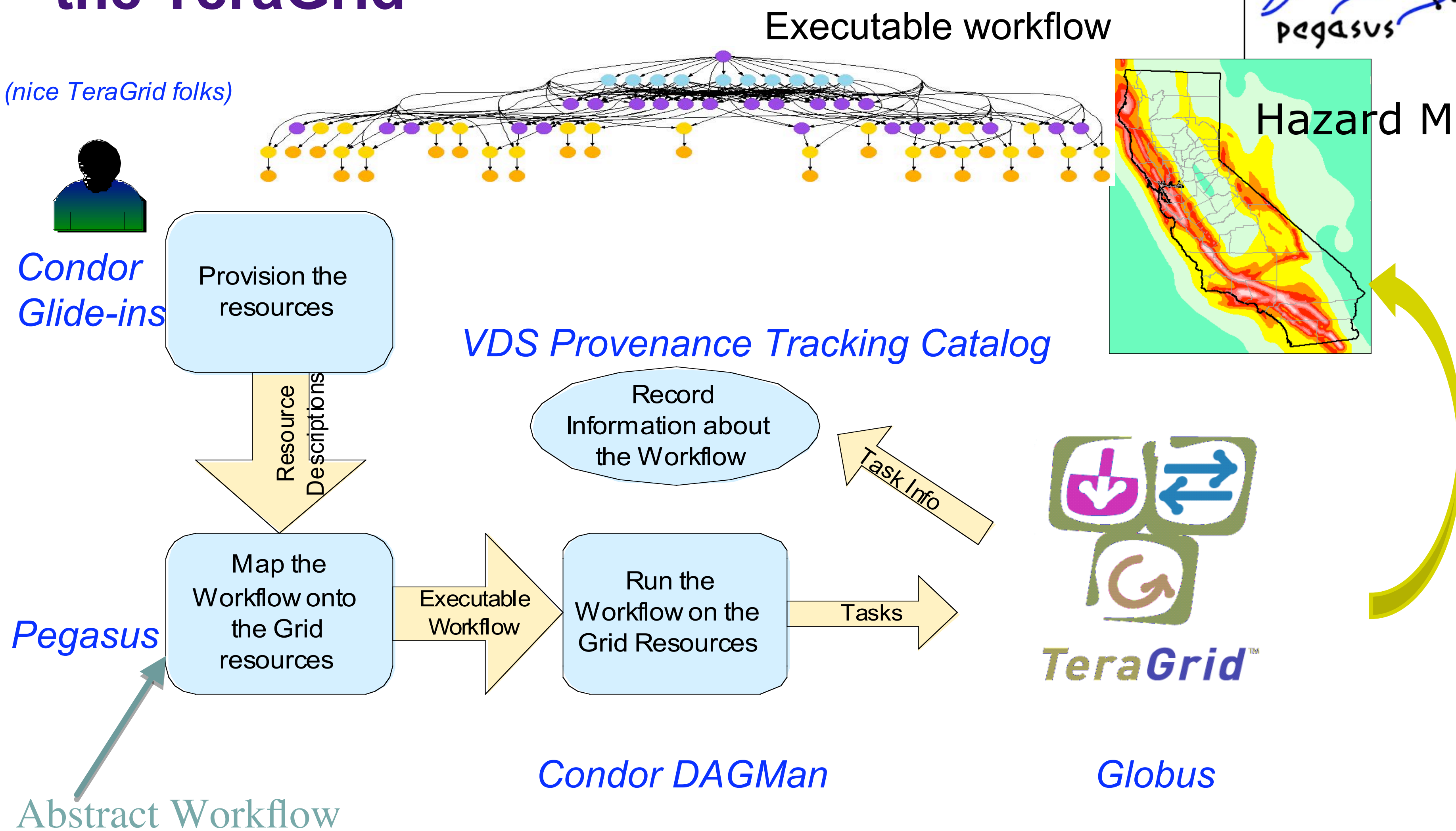


Small 1,200 Montage Workflow

Montage application
~7,000 compute jobs in instance
~10,000 nodes in the executable workflow
same number of clusters as processors
speedup of ~15 on 32 processors

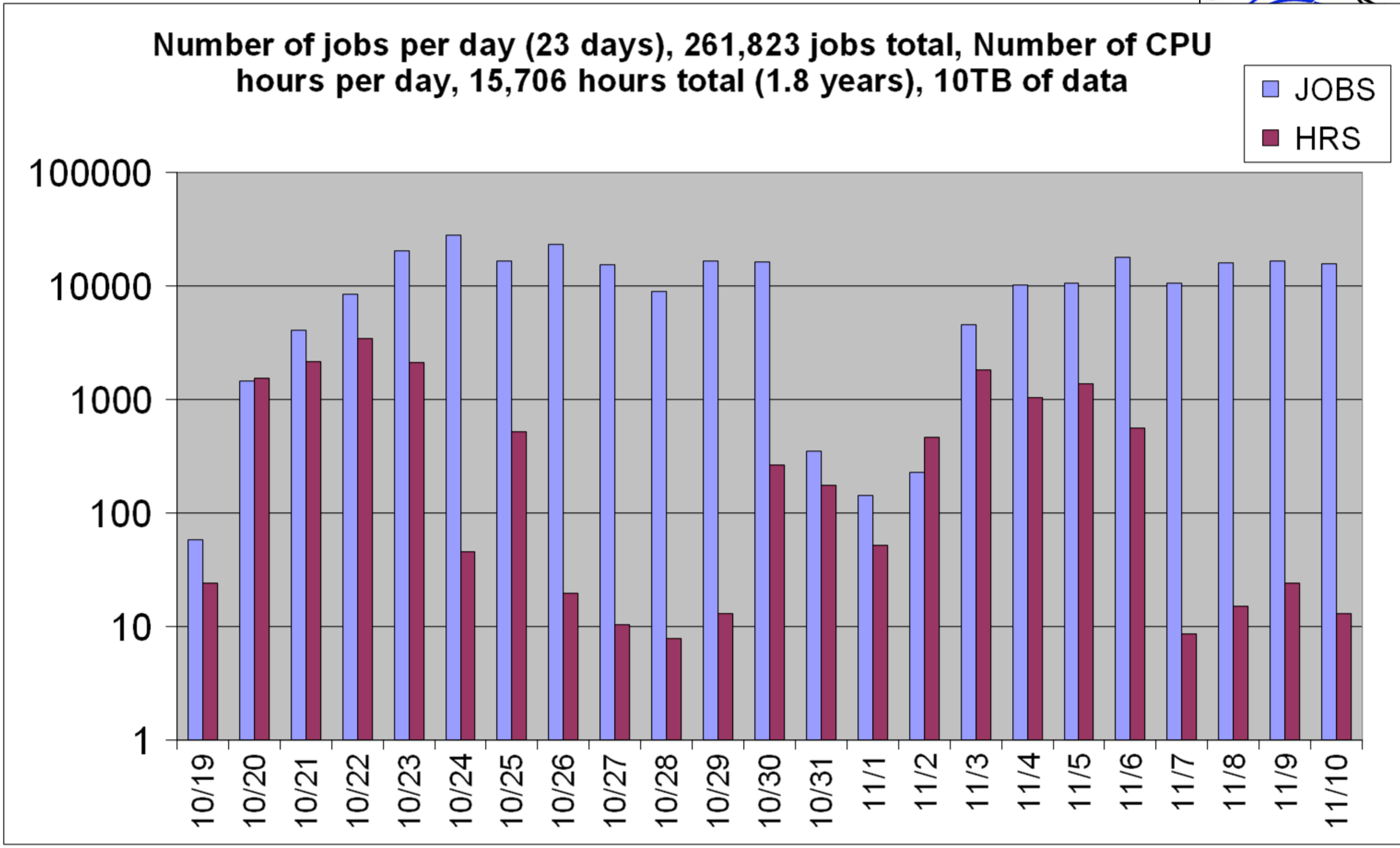


Southern California Earthquake Center (SCEC) provisioning for workflows on the TeraGrid



Joint work with: R. Graves, T. Jordan, C. Kesselman, P. Maechling, D. Okaya & others

Performance results for 2 SCEC sites (Pasadena and USC) on the TeraGrid

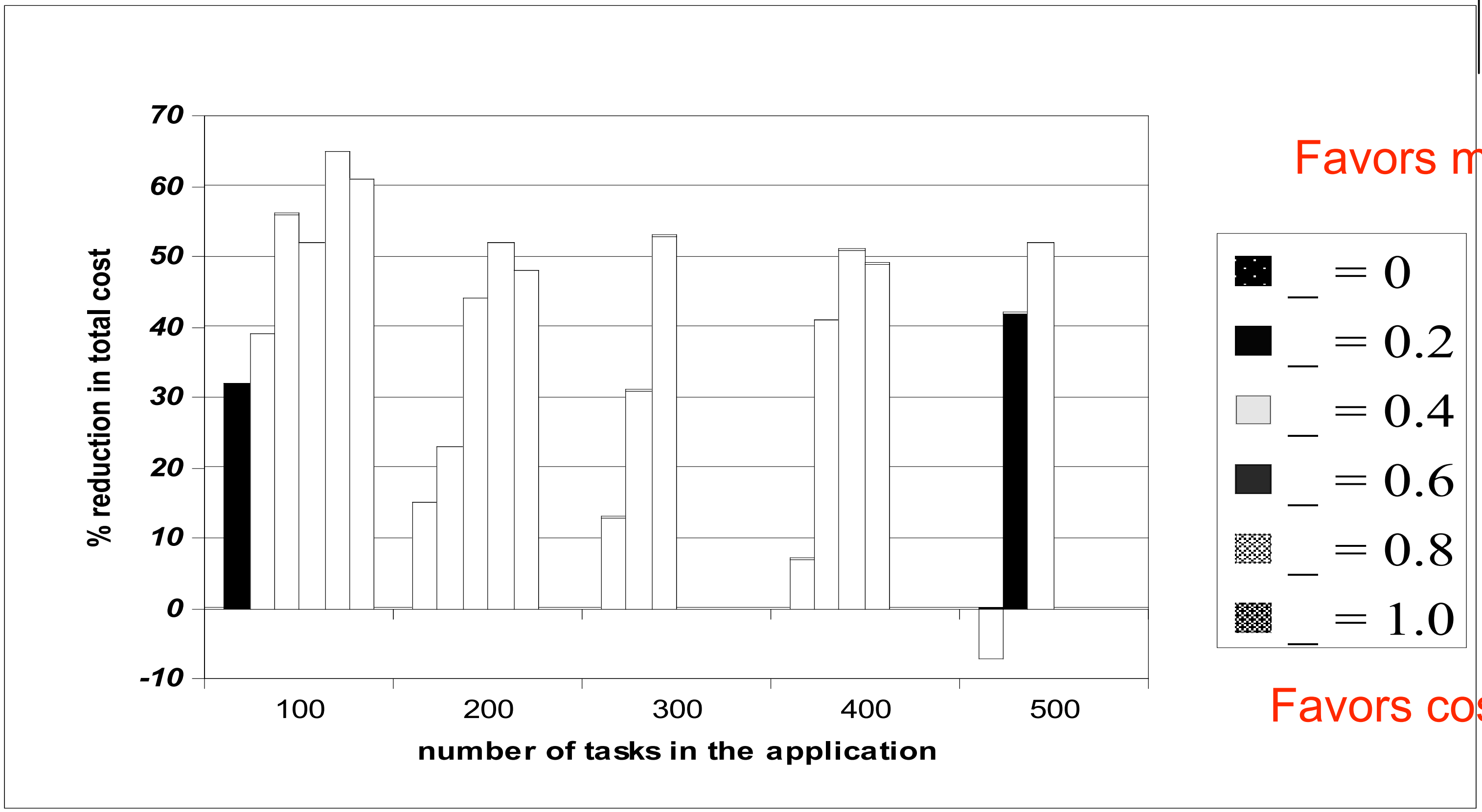


E. Deelman, et al. "Managing Large-Scale Workflow Execution from Resource Provisioning to Provenance tracking: The CyberShake Example", eScience 2006, Amsterdam, December 2006, to appear.

Approach to Provisioning Resources Ahead of the Execution

- Assume resources publish their availability in the form of “slot”
- Pick the slots that would
  - Minimize the workflow makespan, and
  - Minimize the cost of the allocation (proportional to allocation size)
    - Initially slots are indivisible
- Evaluate using Min-min for choosing the slots and Genetic-type algorithms
- Evaluate using random workflows

% reduction in total cost (combines makespan and allocation costs)



Favors makespan

Favors cost of allocations

4 compute sites, ~ 100 processors total, ~200 slots

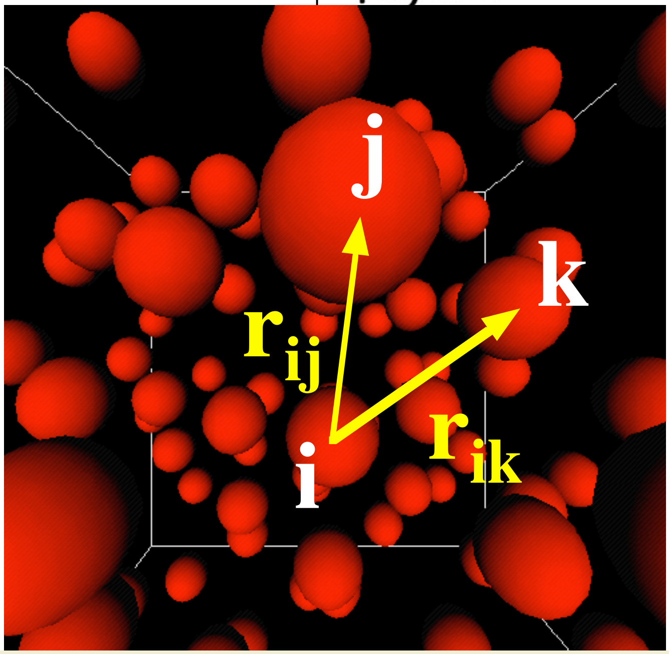
GA in general achieves a 25-30% reduction in the total cost over Min-Min

In 30% of cases, Min-Min could not complete the schedule

G. Singh, C. Kesselman, E. Deelman, Application-level Resource Provisioning on the Grid, e-Science 2006, to appear

Optimizing performance in the large and in the small

- A systematic strategy for composing application components into workflows
- Search for the most appropriate implementation of both components and workflows
- Component optimization
  - Select among implementation ***variants*** of the same computation
  - Derive integer values of optimization ***parameters***
  - Only search promising code variants and a restricted parameter space
- Workflow optimization
  - Knowledge-rich representation of components and workflow properties



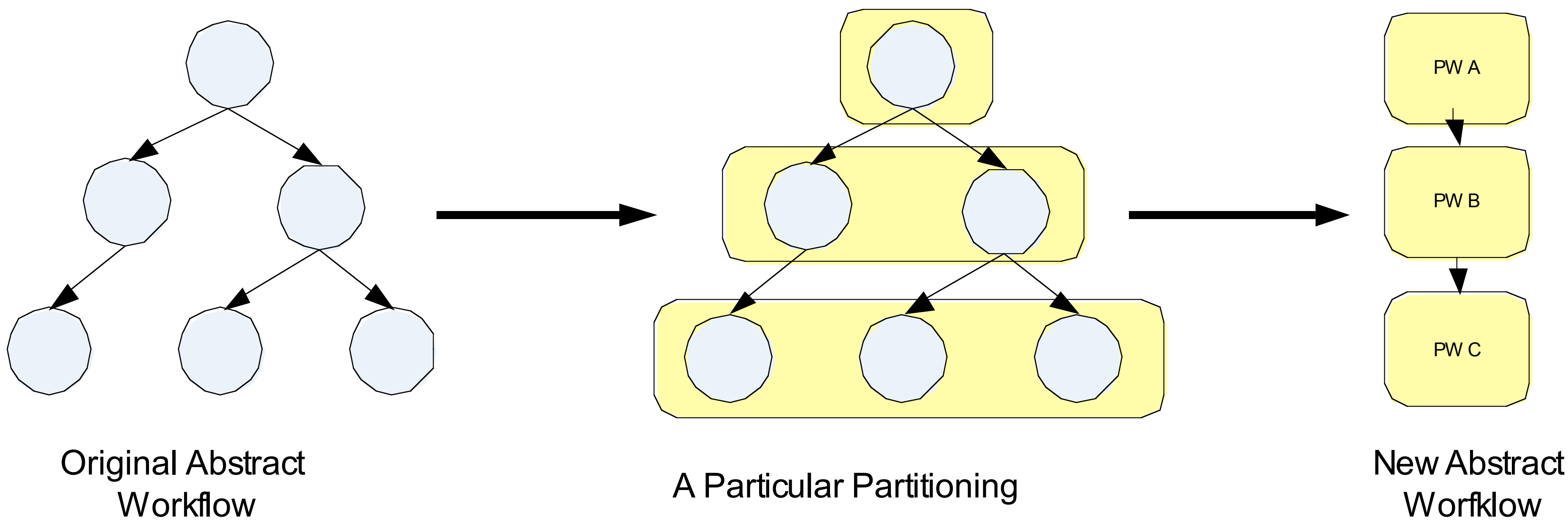
Molecular dynamics application
Aiichiro Nakano, Ashish Sharma, (USC, OSU)

"A Systematic Approach to Composing and Optimizing Application Workflows," E. Deelman, M. Hall, Y. Gil, K. Lerman, and J. Saltz, In *Proceedings of the Workshop on Patterns in High Performance Computing*, May, 2005.

Outline

- Pegasus
- Challenges in Workflow Performance
  - Workflow restructuring
  - Provisioning resources
  - Modeling and optimizing workflow component behavior
- Challenges in Workflow Reliability
  - Mapping portions of the workflow at a time
  - Efficient data handling
- Providing workflow mapping capabilities to a variety of workflow generation mechanism
- Application Experiences and Science Impacts
- Conclusions

Managing execution environment changes through partitioning

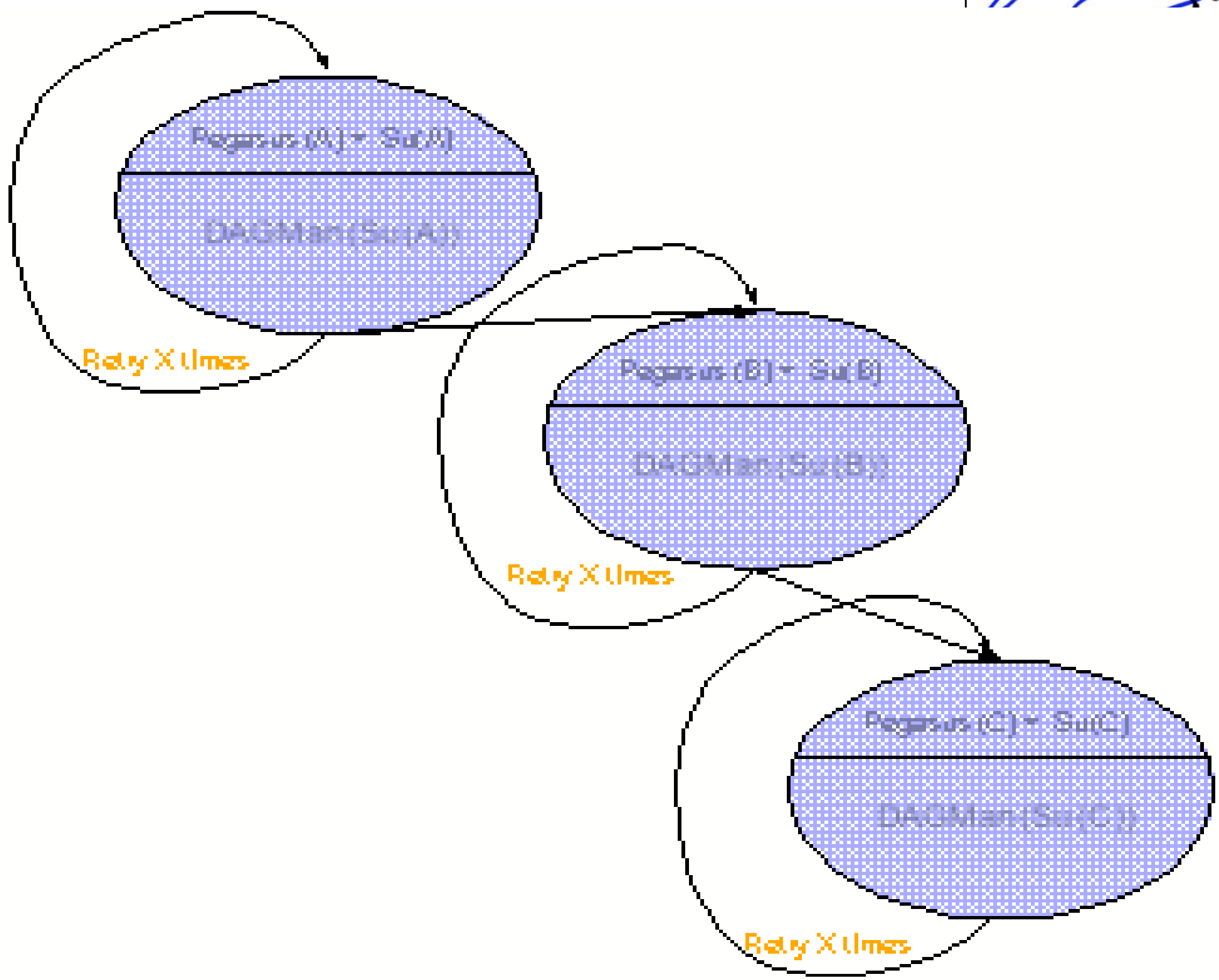


E. Deelman, et al. Pegasus: a Framework for Mapping Complex Scientific Workflows onto Distributed Systems, Scientific Programming Journal, Volume 13, Number 3, 2005

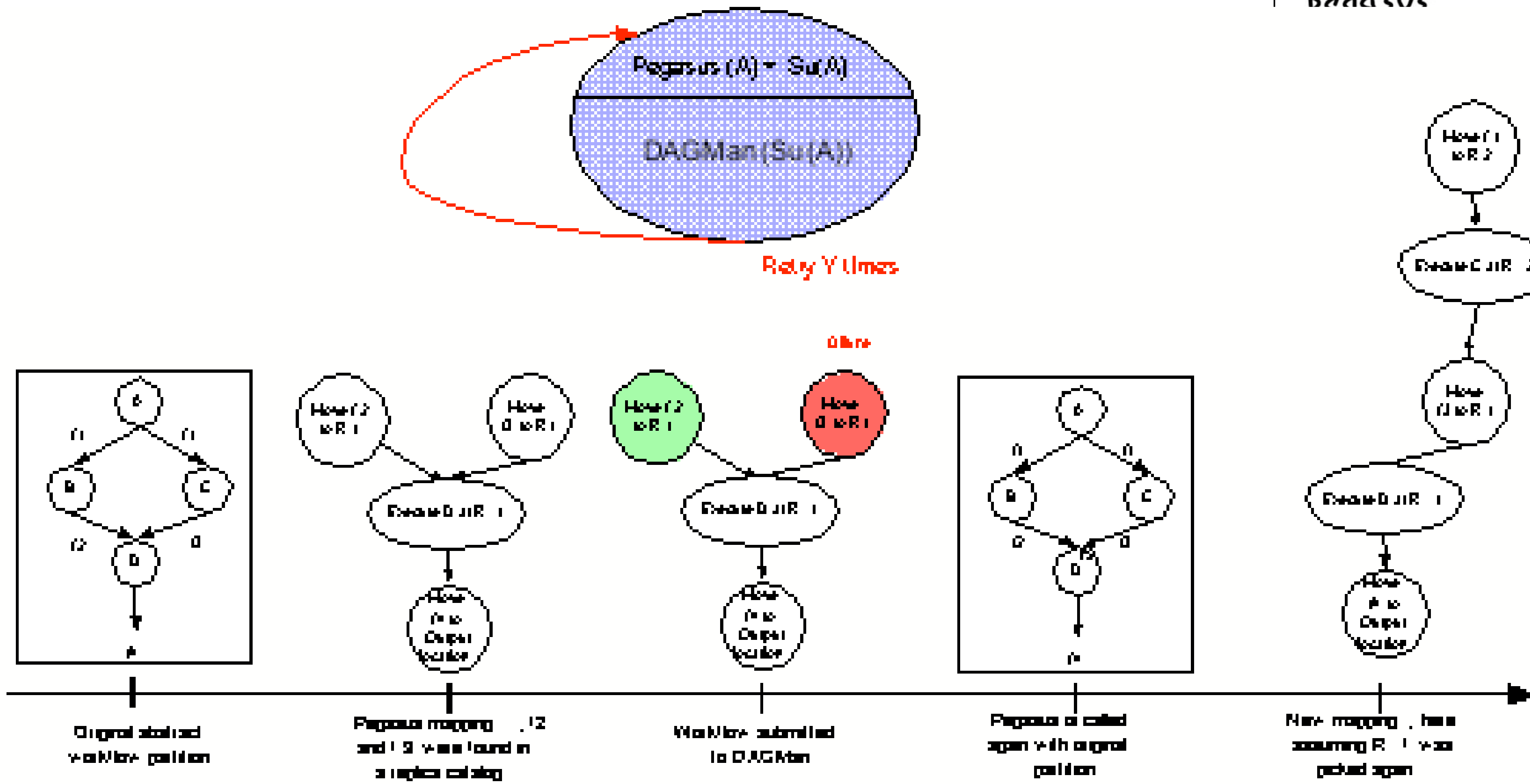
Resulting Meta-Workflow

Pegasus (X): Pegasus generates the concrete workflow and the submit files for Partition X -- Su(X)

DAGMan(Su(X): DAGMan executes the concrete workflow for X



Re-mapping in case of failures



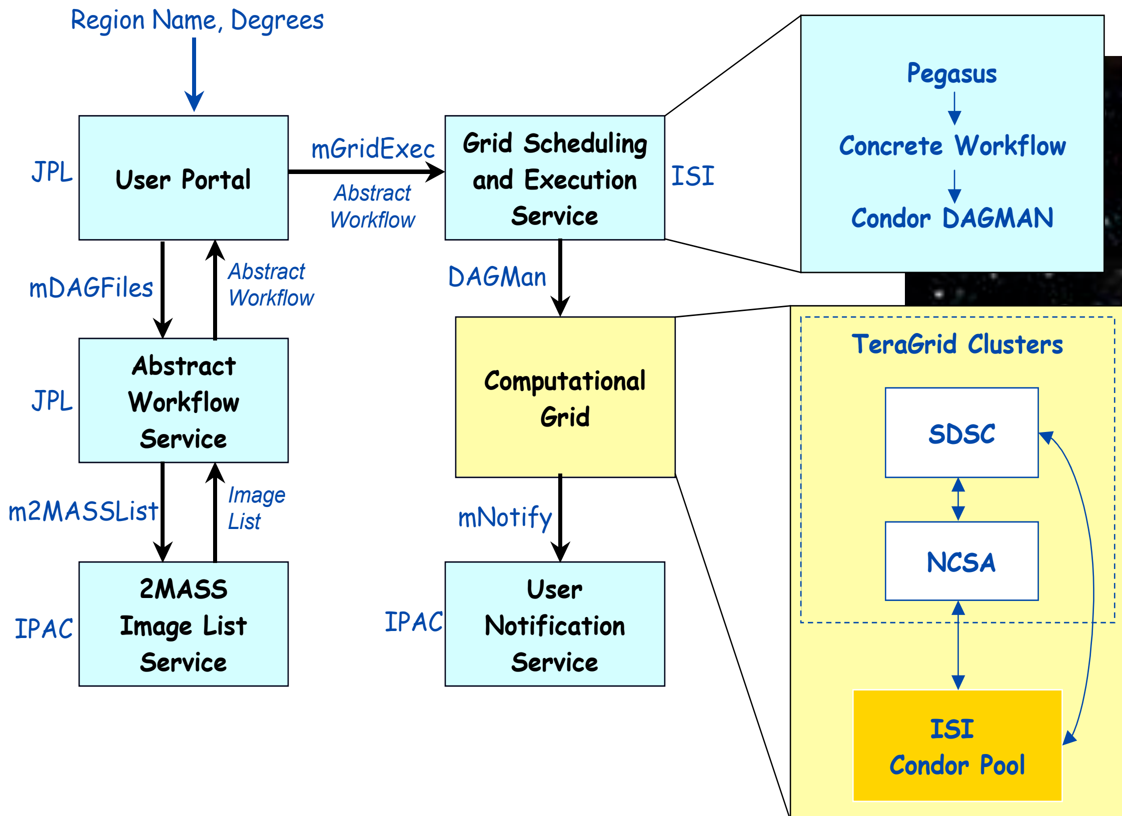
Efficient data handling

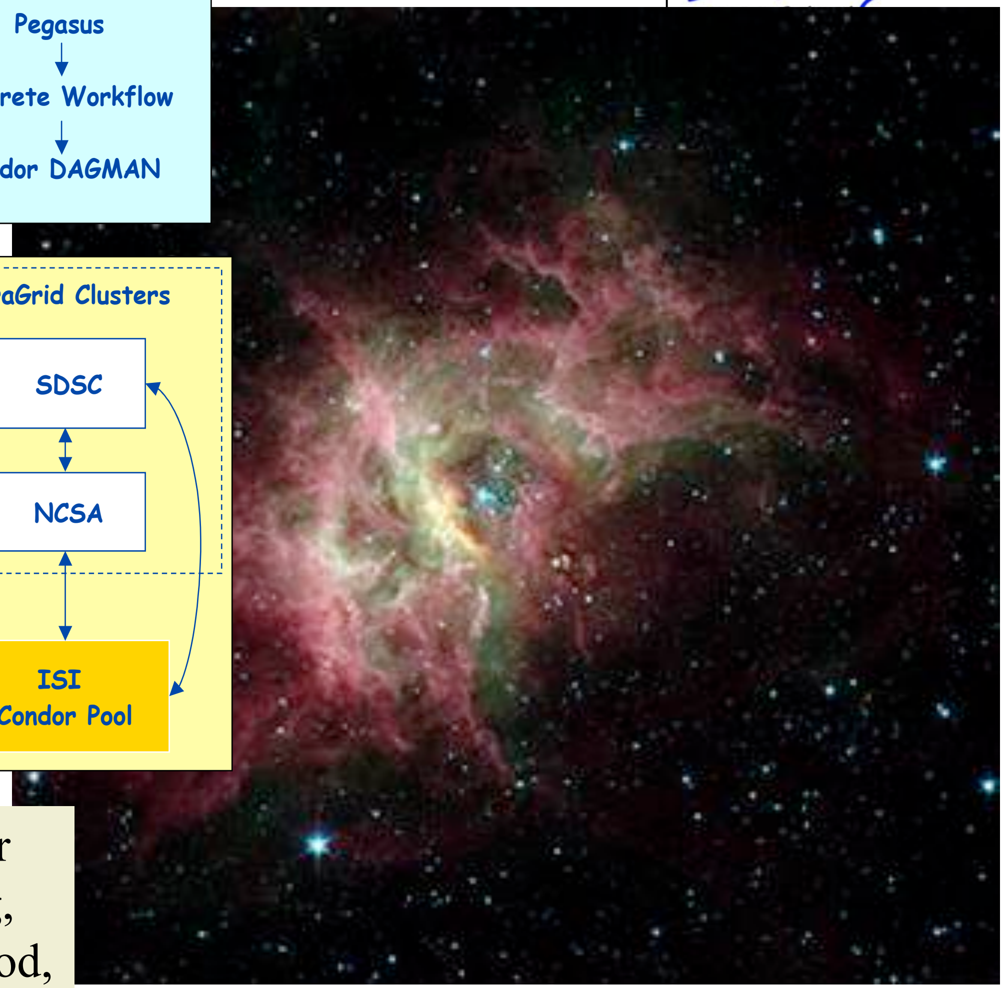
- Input data is staged dynamically
- New data products are generated during execution
- For large workflows 10,000+ files
  - Similar order of intermediate and output files
  - Total space occupied is far greater than available space—failures occur
- Solution:
  - Determine which data is no longer needed and when
  - Add nodes to the workflow do cleanup data along the way
- Issues:
  - minimize the number of nodes and dependencies added so as not to slow down workflow execution
  - deal with portions of workflows scheduled to multiple sites
  - deal with files on partition boundaries

Outline

- Pegasus
- Challenges in Workflow Performance
  - Workflow restructuring
  - Provisioning resources
  - Modeling and optimizing workflow component behavior
- Challenges in Workflow Reliability
  - Mapping portions of the workflow at a time
  - Efficient data handling
- Providing workflow mapping capabilities to a variety of workflow generation mechanism
- Application Experiences and Science Impacts
- Conclusions

Portals, Providing high-level Interfaces

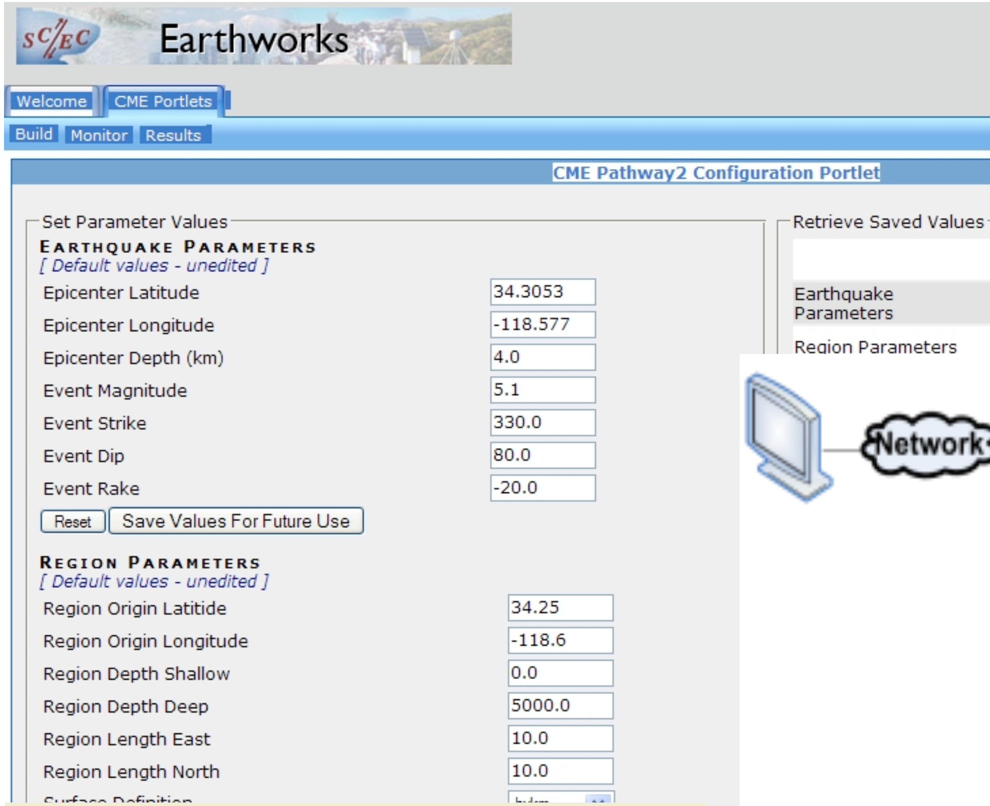




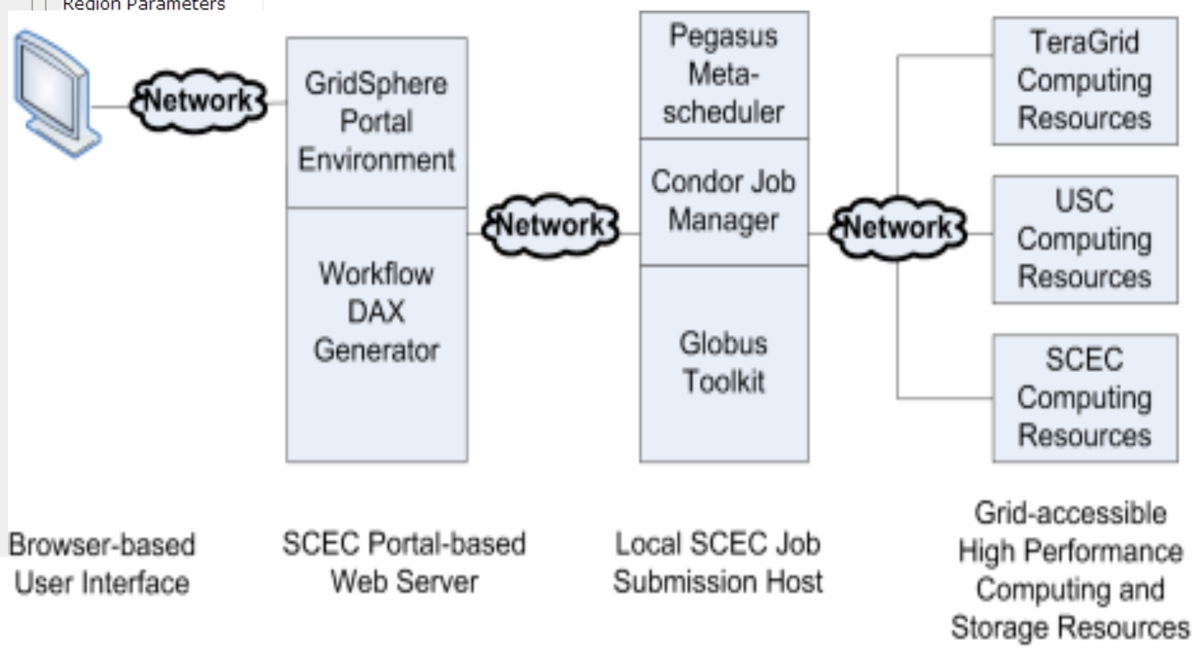
Montage: a grid portal and software toolkit for science-grade astronomical image mosaicking, J. C. Jacob, D. S. Katz, G. B. Berriman, J. Good, A. C. Laity, E. Deelman, C. Kesselman, G. Singh, M.-H. Su, T. A. Prince, R. Williams, , IJCSE, *to appear 2006*

Galactic Star Formation Region RCW 49

Portals, Providing high-level Interfaces



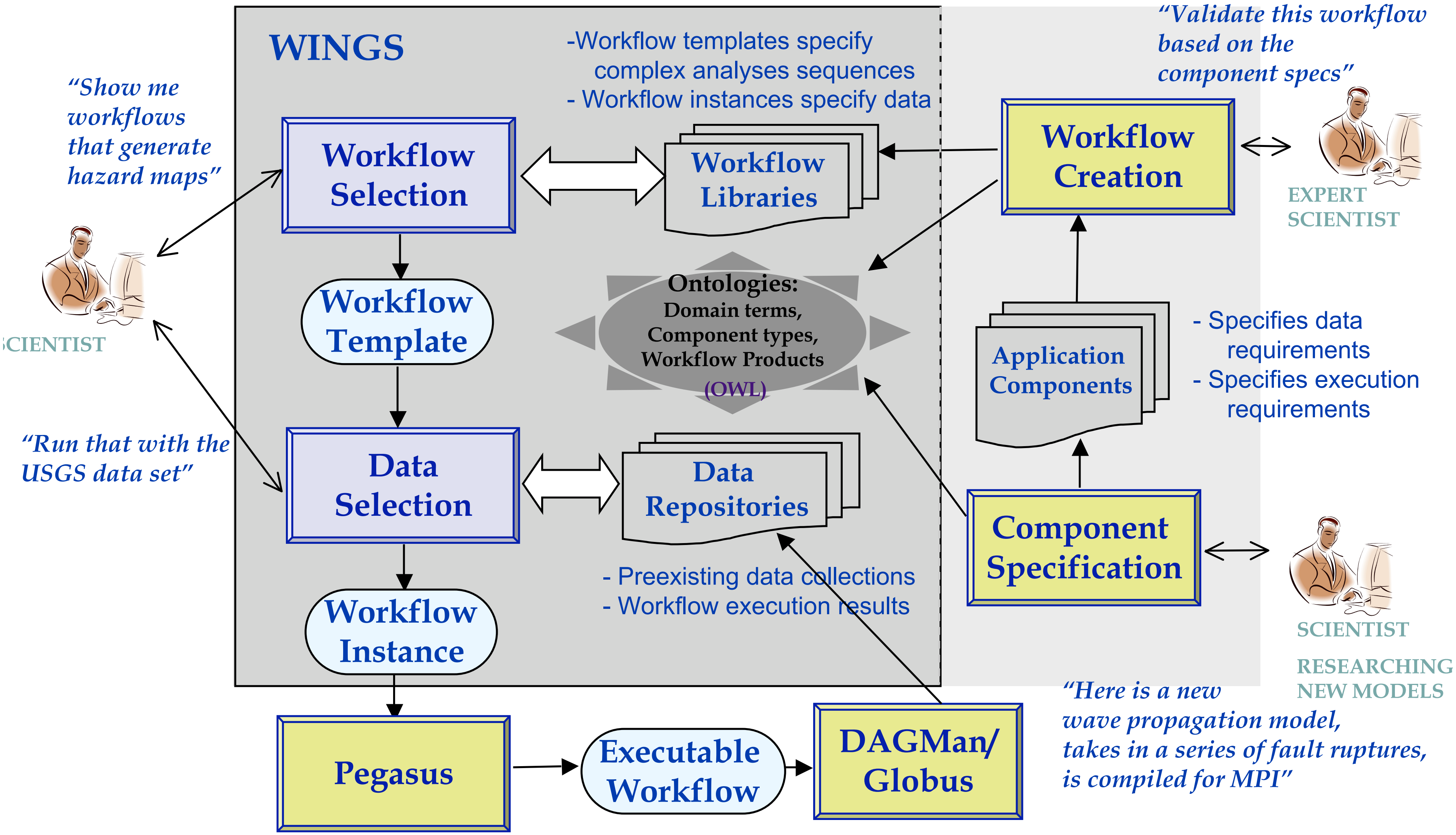
TG Science Gateway, Washington University



EarthWorks Project (SCEC), lead by with J. Muench P. Maechling, H. Francoeur, and others

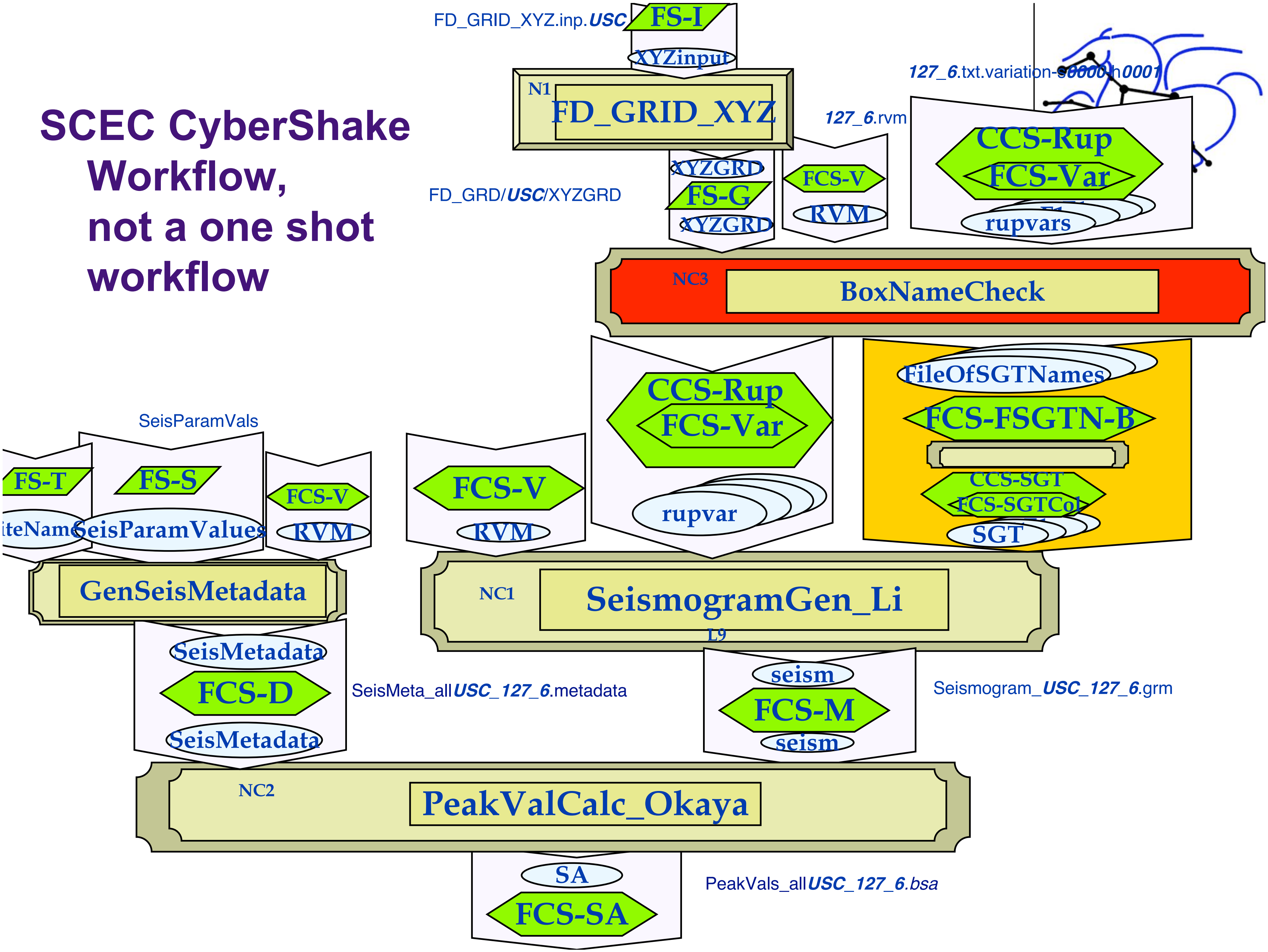
SCEC Earthworks: Community Access to Wave Propagation Simulations, J. Muench, H. Francoeur, D. Okaya, Y. Cui, P. Maechling, E. Deelman, G. Mehta, T. Jordan TG 2006

WINGS/Pegasus: Workflow Instance Generation and Selection, Using semantic technologies for workflow generation

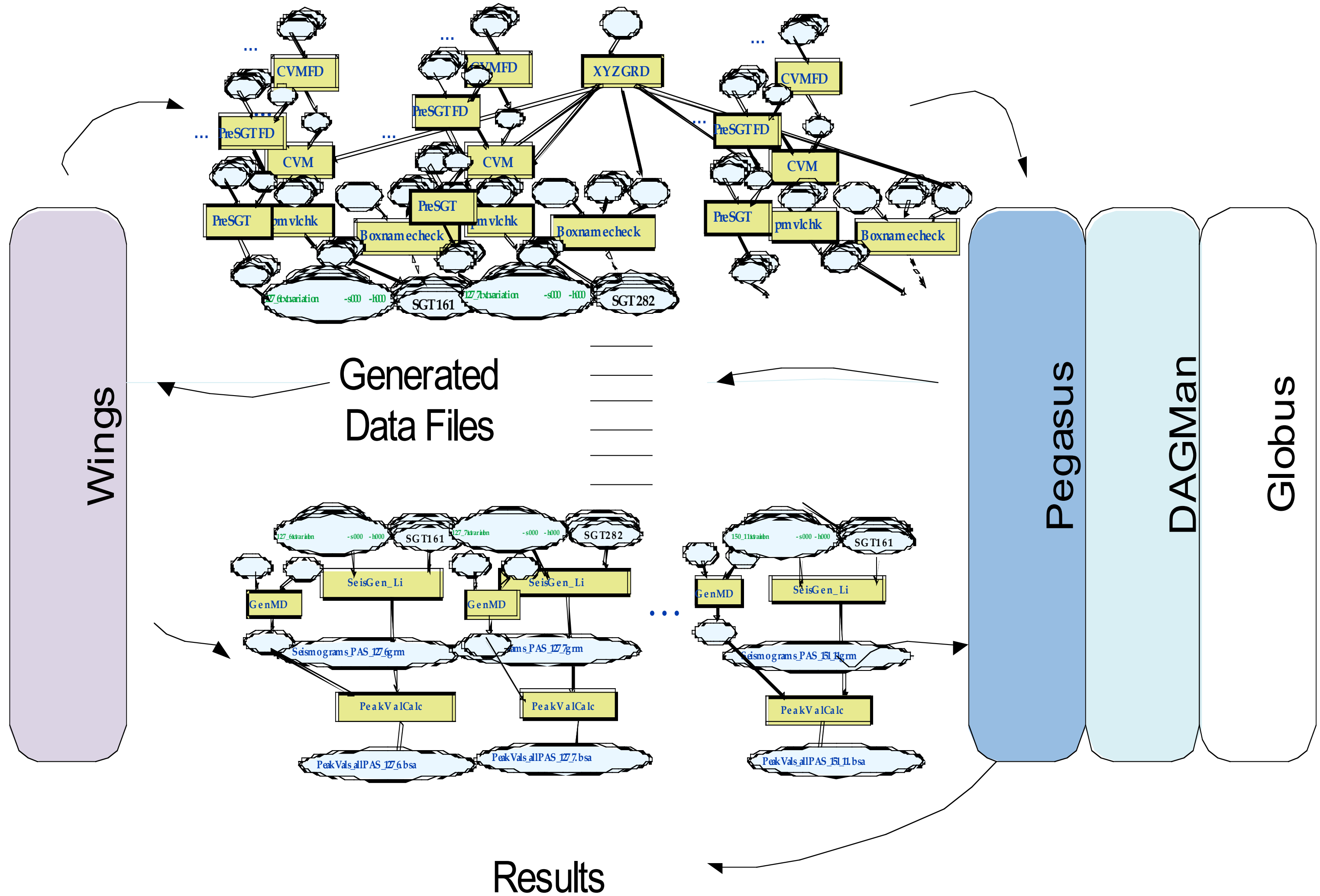


Wings for Pegasus: A Semantic Approach to Creating Very Large Scientific Workflows
Yolanda Gil, Varun Ratnakar, Ewa Deelman, Marc Spraragen, and Jihie Kim, *OWL: Experiences and Directions 2006*

SCEC CyberShake Workflow, not a one shot workflow



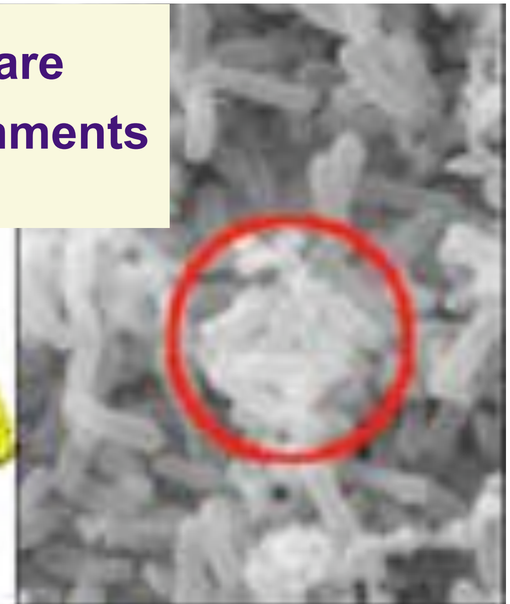
Iterative workflow instantiation, mapping and execution

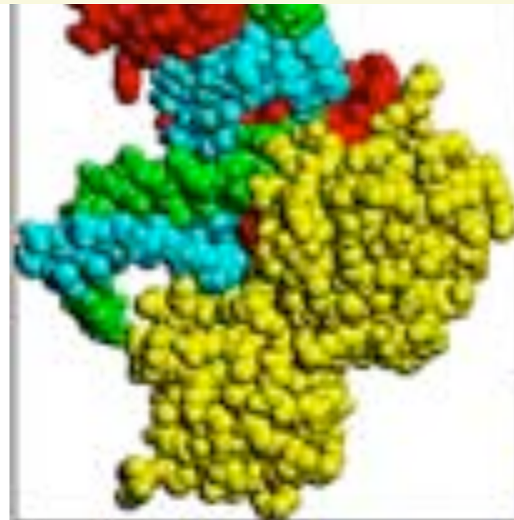


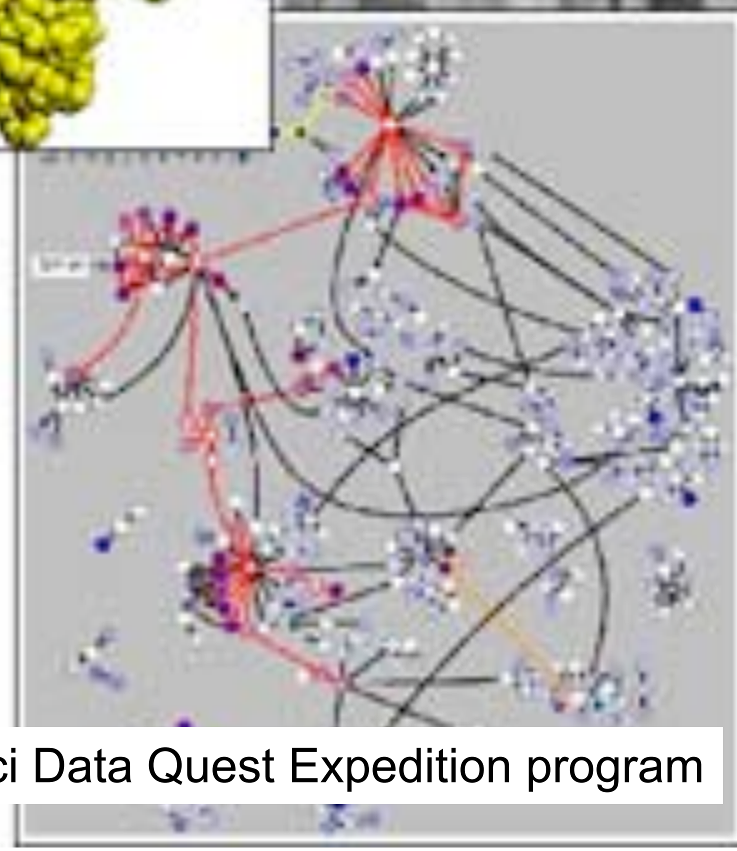
Outline

- Pegasus
- Challenges in Workflow Performance
  - Workflow restructuring
  - Provisioning resources
  - Modeling and optimizing workflow component behavior
- Challenges in Workflow Reliability
  - Mapping portions of the workflow at a time
  - Efficient data handling
- Providing workflow mapping capabilities to a variety of workflow generation mechanism
- Application Experiences and Science Impacts
- Conclusions

BLAST: set of sequence comparison algorithms that are used to search sequence databases for optimal local alignments to a query







2 major runs were mapped using Pegasus and related technologies

1) 60 genomes (4,000 sequences each),

In 24 hours processed Genomes selected from DOE-sponsored sequencing projects

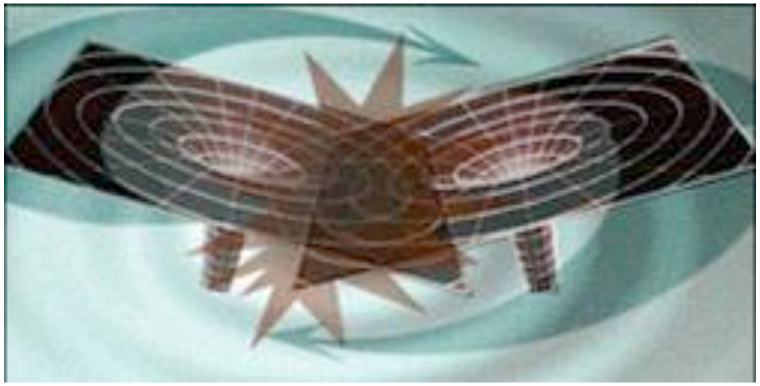
- 67 CPU-days of processing time delivered
- ~ 10,000 Grid jobs
- >200,000 BLAST executions
- 50 GB of data generated

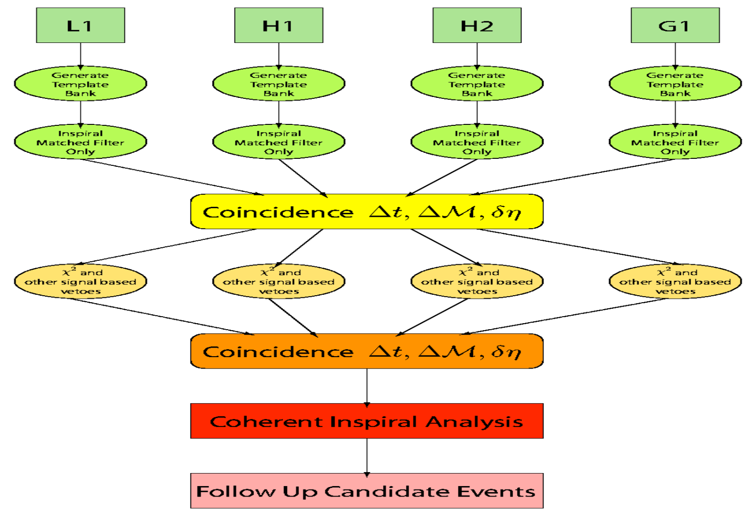
2) 450 genomes processed

Speedup of 5-20 times were achieved because the compute nodes we used efficiently by keeping the submission of the jobs to the compute cluster constant.

Lead by Veronika Nefedova (ANL) as part of the Paci Data Quest Expedition program

Laser Gravitational Wave Observatory

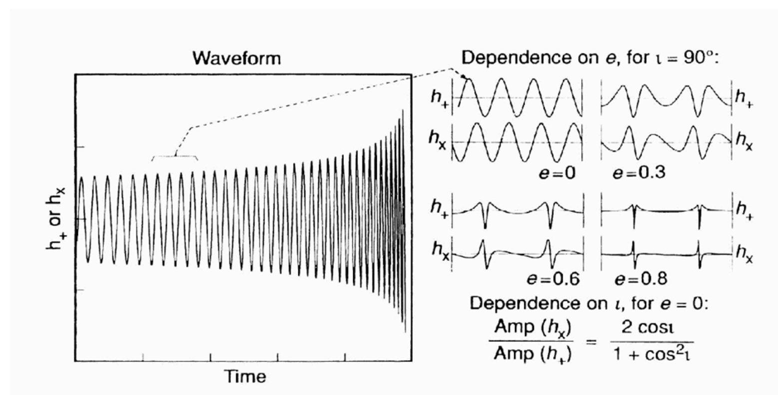




LIGO's Binary Inspiral Analysis

- Based on the precise comparison of known waveforms from the final moments of orbital evolution in a system of two neutron stars or black holes

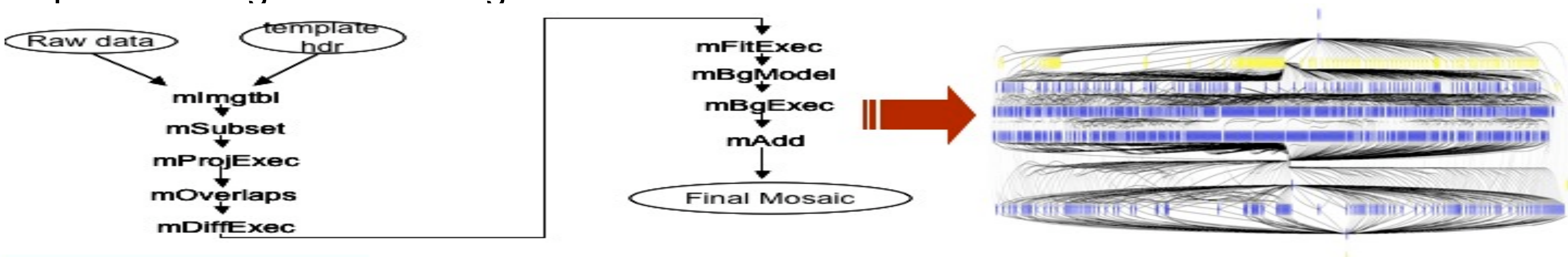
- Pegasus runs large-scale LIGO workflows on the Open Science Grid
- A month of LIGO data requires many thousands of jobs, running for days on hundreds of CPUs

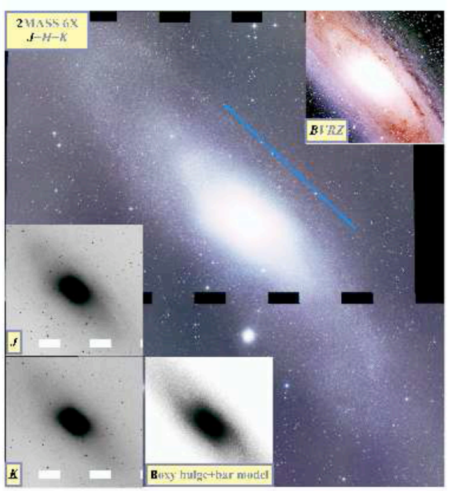


LIGO OSG effort is led by Kent Blackburn and David Meyers (Caltech)

National Virtual Observatory and Montage: Building Science-Grade Mosaics of the Sky

Workflow technologies were used to transform a single-processor code into a complex workflow and parallelized computations to process larger-scale images.





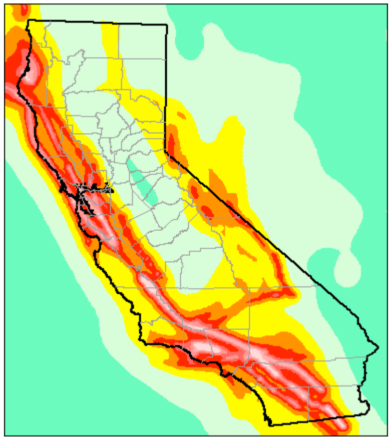
- Pegasus maps workflows with thousands of tasks onto NSF's TeraGrid
- Pegasus improved overall runtime by 90% through automatic workflow restructuring and minimizing execution overhead

Montage Science Result : Verification of a Bar in the Spiral Galaxy M31, Beaton et al. *Ap J Lett* in press

Eleven major projects and surveys world wide, such as the Spitzer Space Telescope Legacy teams have integrated Montage into their pipelines and processing environments to generate science and browse products for dissemination to the astronomy community.

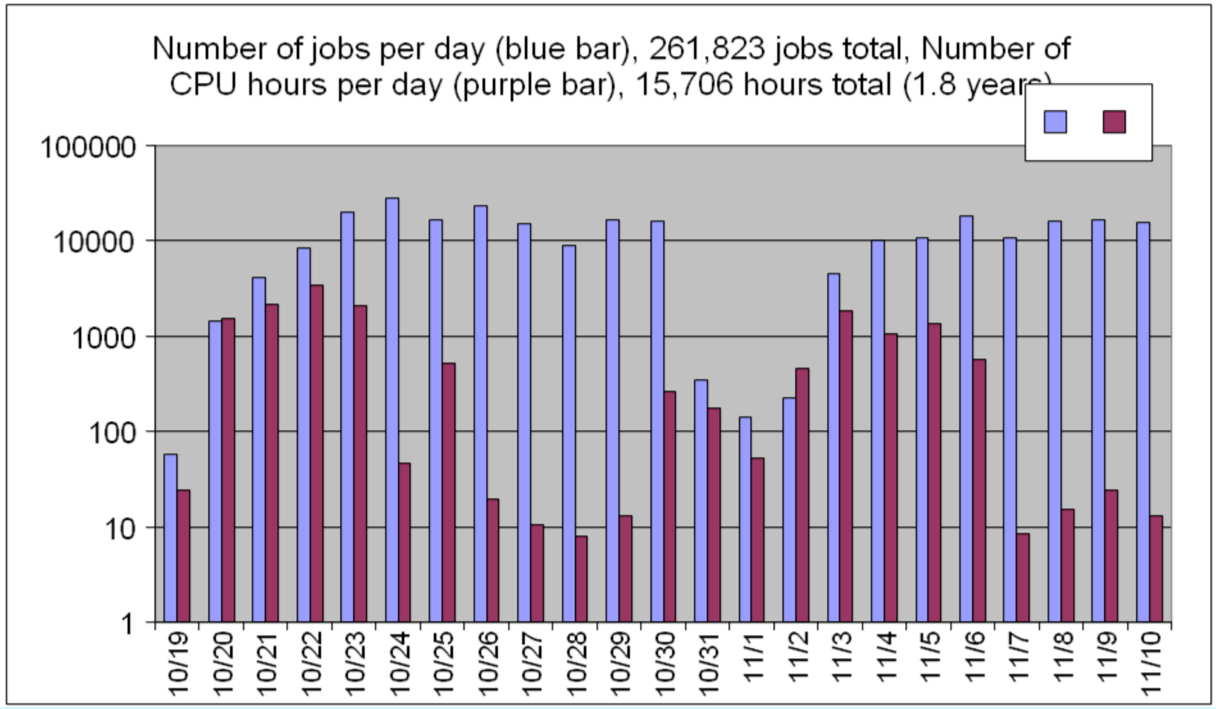
Montage is a collaboration between IPAC, JPL and CACR

Southern California Earthquake Center (SCEC)



- SCEC's Cybershake is used to create Hazard Maps that specify the maximum shaking expected over a long period of time
- Used by civil engineers to determine building design tolerances

Pegasus mapped SCEC CyberShake workflows onto the TeraGrid in Fall 2005. The workflows ran over a period of 23 days and processed 20TB of data using 1.8 CPU Years. Total tasks in all workflows: 261,823.



CyberShake Science result*: CyberShake delivers new insights into how rupture directivity and sedimentary basin effects contribute to the shaking experienced at different geographic locations. As a result more accurate hazard maps can be created.*

SCEC is led by Tom Jordan, USC

Benefits of Scientific Workflows (from the point of view of an application scientist)

- Conducts a series of computational tasks.
  - Resources distributed across Internet.
- Chaining (outputs become inputs) replaces manual hand-offs.
  - Accelerated creation of products.
- Ease of use - gives non-developers access to sophisticated codes.
  - Avoids need to download-install-learn how to use someone else's code.
- Provides framework to host or assemble community set of applications.
  - Honors original codes. Allows for heterogeneous coding styles.
- Framework to define common formats or standards when useful.
  - Promotes exchange of data, products, codes. Community metadata.
- Multi-disciplinary workflows can promote even broader collaborations.
  - E.g., ground motions fed into simulation of building shaking.
- Certain rules or guidelines make it easier to add a code into a workflow.

Slide courtesy of David Okaya, SCEC, USC

Outline

- Pegasus
- Challenges in Workflow Performance
  - Workflow restructuring
  - Provisioning resources
  - Modeling and optimizing workflow component behavior
- Challenges in Workflow Reliability
  - Mapping portions of the workflow at a time
  - Efficient data handling
- Providing workflow mapping capabilities to a variety of workflow generation mechanism
- Application Experiences and Science Impacts
- Conclusions

Pegasus: Planning for Execution in Grids

- Pegasus bridges the scientific domain and the execution environment
- Pegasus enables scientists to construct workflows in abstract terms without worrying about the details of the underlying CyberInfrastructure
- Pegasus is used day-to-day to map complex, large-scale scientific workflows with thousands of tasks processing TeraBytes of data
- Pegasus applications include NVO's Montage, SCEC's CyberShake simulations, LIGO's Binary Inspiral Analysis, and others
- Pegasus improves the performance of applications through:
  - Data reuse to avoid duplicate computations and provide reliability
  - Workflow restructuring to improve resource allocation
  - Automated task and data transfer scheduling to improve overall runtime
- Pegasus provides reliability through dynamic workflow remapping
- Pegasus uses Condor's DAGMan for workflow execution and Globus to provide the middleware for distributed environments

Current and Future Research

- Resource selection
- Resource provisioning
- Workflow restructuring
- Adaptive computing
  - Workflow refinement adapts to changing execution environment
- Workflow provenance (including provenance of the mapping process) – new collaboration with Luc Moreau
- Management and optimization across multiple workflows
- Workflow debugging
- Streaming data workflows
- Automated guidance for workflow restructuring
- Support for long-lived and recurrent workflows

Mosaic of M42 created on the Teragrid resources using Pegasus

Pegasus improved the runtime of this application by 90% over the baseline case, creating as many clusters as available processors

Workflow with 4,500 nodes

Bruce Berriman, John Good (Caltech)
Joe Jacob, Dan Katz (JPL)
Gurmeet Singh, Mei Su (ISI)

Montage mar

Scientific Workflows

- Current workflow approaches are exploring specific aspects of the problem:
  - Creation, reuse, provenance, performance, reliability
- New requirements are emerging
  - Streaming data, from batch to interactive steering, event-driven analysis, collaborative design of workflows
- Need to develop a science of workflows
  - A more comprehensive treatment of workflow lifecycle
  - Understand current and long-term requirements from science applications
    - reproducibility
  - Workflows as first-class citizens in CyberInfrastructure

Acknowledgments

- The Pegasus team consists of Ewa Deelman, Gaurang Mehta, Mei-Hui Su, and Karan Vahi (ISI)
- Thanks to Yolanda Gil (ISI) for collaboration on scientific workflow issues
- Thanks to Montage collaborators: Bruce Berriman, John Good, Dan Katz, and Joe Jacobs
- Thanks to SCEC collaborators: Tom Jordan, Robert Graves, Phil Maechling, David Okaya, Li Zhao
- Thanks to LIGO collaborators: Kent Blackburn, Duncan Brown, and David Meyers
- Thanks to the National Science Foundation for the support of this work

Relevant Links

- Pegasus: pegasus.isi.edu
  - released as part of VDS, joint work with Ian Foster
- NSF Workshop on Challenges of Scientific Workflows: vtcpc.isi.edu/wiki/, E. Deelman and Y. Gil (chairs)
- Workflows for e-Science, Taylor, I.J.; Deelman, E.; Gannon, D.B.; Shields, M. (Eds.), Dec. 2006, *to appear*
- Wings: www.isi.edu/ikcap/wings/
- SCEC: www.scec.org
- Montage: montage.ipac.caltech.edu/
- LIGO: www.ligo.caltech.edu/
- Globus: www.globus.org
- Condor: www.cs.wisc.edu/condor/
- TeraGrid: www.teragrid.org
- Open Science Grid: www.opensciencegrid.org

Benefits of the workflow & Pegasus approach

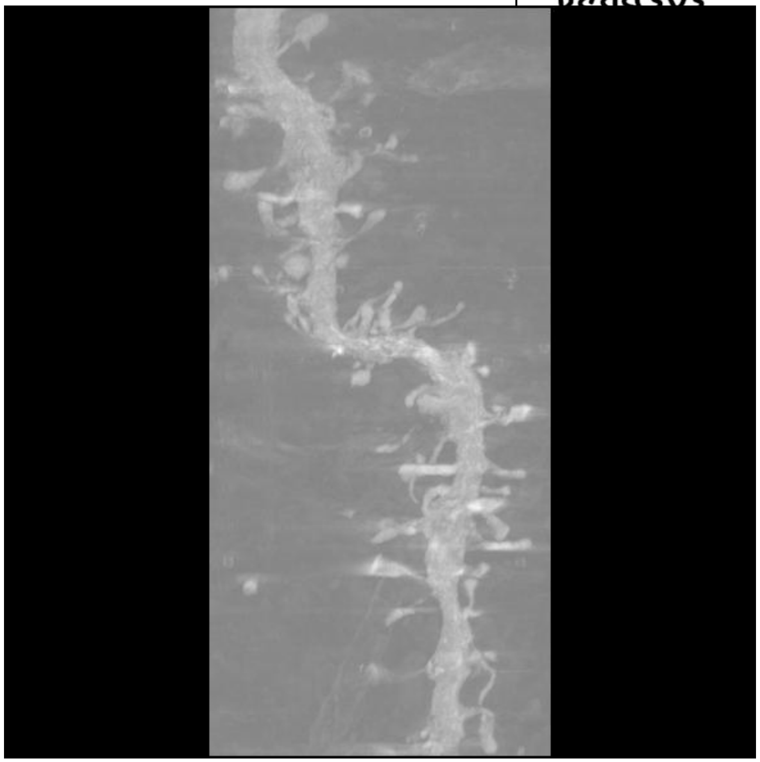
- Pegasus can run the workflow on a variety of resources
- Pegasus can run a single workflow across multiple resources
- Pegasus can opportunistically take advantage of available resources (through dynamic workflow mapping)
- Pegasus can take advantage of pre-existing intermediate data products
- Pegasus can improve the performance of the application.

General Conclusions

- Workflows are recipes for Cyberinfrastructure
- Need to support the dynamic nature of science
- Support for long-lived and recurrent workflows
- Many challenges and many workflow tools out there
  - Interoperability is desired
- Need common representations that can be used by various workflow management systems
  - Maybe semantic technologies?
- Need common provenance tracking capabilities
  - See IPAW 06
- To make forward progress, collaboration with application scientists is essential

Tomography (NIH-funded project)

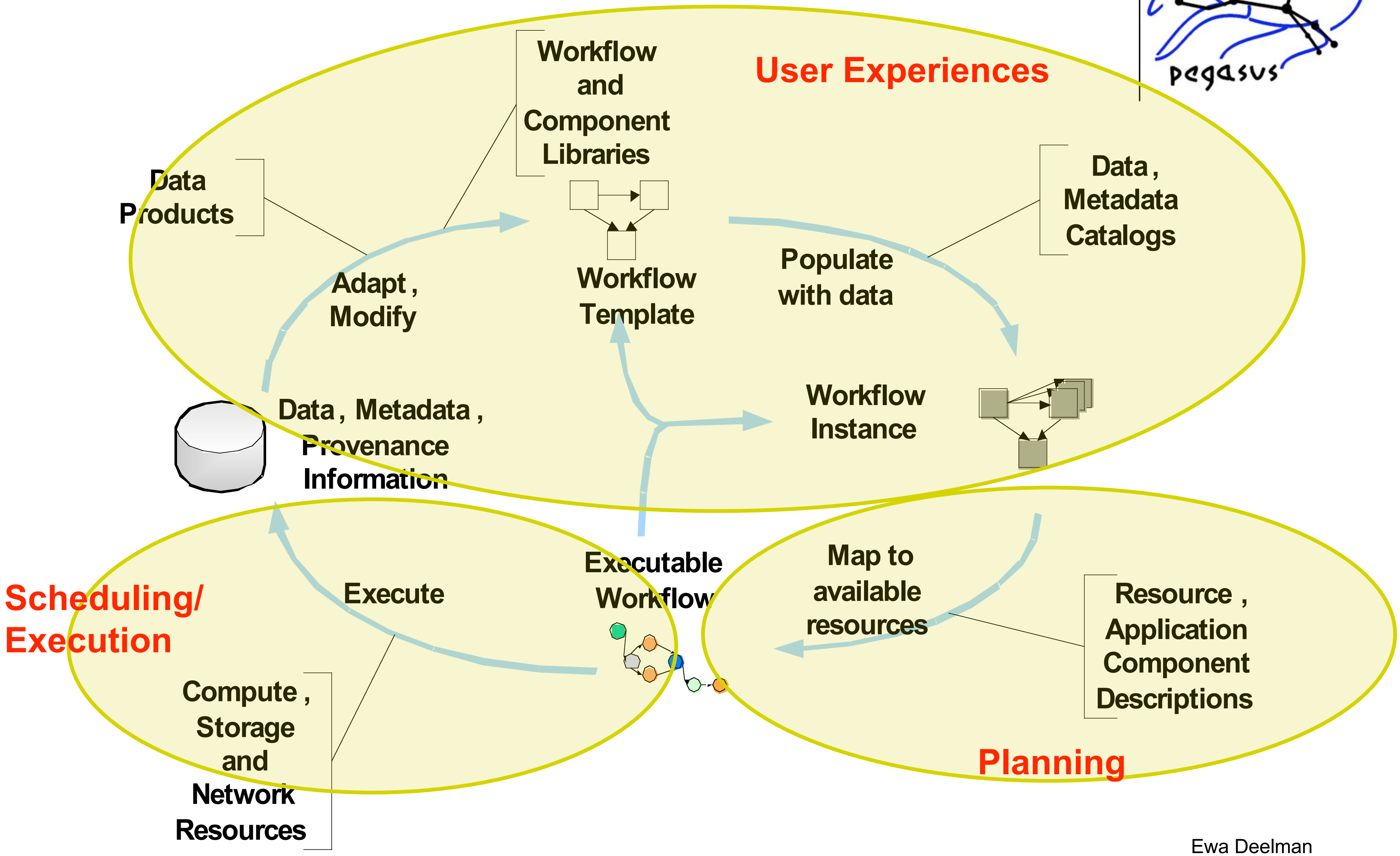
- Derivation of 3D structure from a series of 2D electron microscopic projection images,
- Reconstruction and detailed structural analysis
  - complex structures like synapses
  - large structures like dendritic spines.
- Acquisition and generation of huge amounts of data
- Large amount of state-of-the-art image processing required to segment structures from extraneous background.



Dendrite structure to be rendered by Tomography

Work performed with Mark Ellisman, Steve Peltier, Abel Lin, Thomas Molina (SDSC)

Workflow Lifecycle



Ewa Deelman
www.isi.edu/~deelman

Scientific Workflows

- Emerging paradigm for large-scale and large-scope scientific inquiry
  - Large-scope science integrates diverse models, phenomena, disciplines
- Provide a formalization of the scientific analysis
  - analysis routines to be executed, the data flow amongst them, and relevant execution details
- Provide a systematic way to capture scientific methodology
- Provide provenance information for their results
- Are collaboratively designed, assembled, validated, analyzed
- Should be shared just like today data collections and compute resources are shared among communities
- Used in many scientific disciplines today (SCEC, NVO, LIGO, SEEK, myGrid, many others)

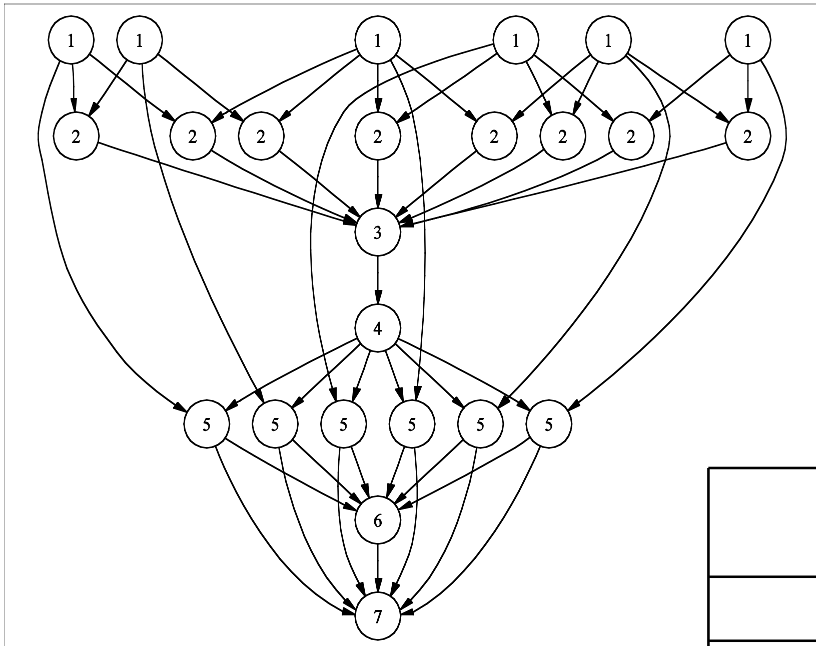
Science Today

- Collaborations of scientists working together towards a common goal
  - Contributing ideas, resources, codes
- Infrastructure that enables the sharing of the resources (compute, data) within a collaboration
  - Globus, Condor
- Technologies that enable the sharing of ideas between scientists
  - Wikis, Email, Access Grid
- Emerging technologies that help share and combine individual software into larger analysis
  - Workflows

Motivation

- Scientists want to describe workflows at a high level without worrying about the execution environment
- There are many distributed resources available to collaborations
- How to efficiently map from the high-level descriptions onto the available resources?
  - While delivering performance and reliability

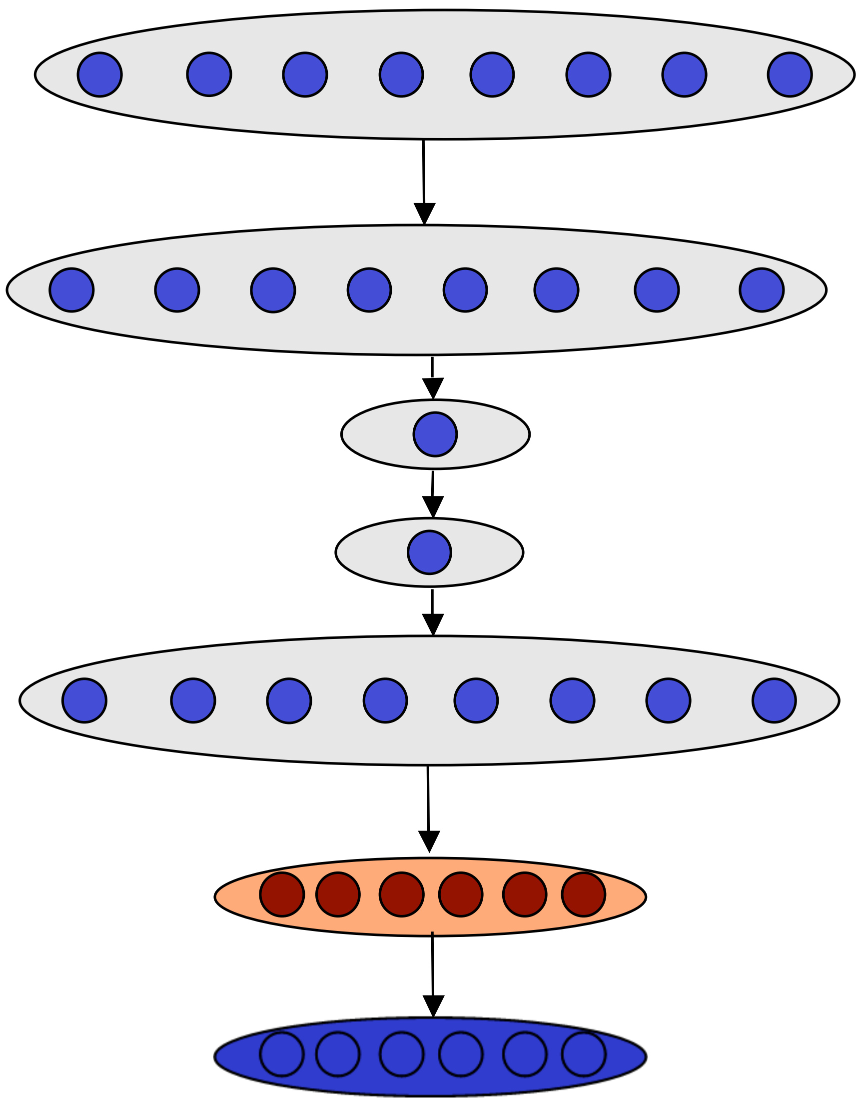
Node Clustering

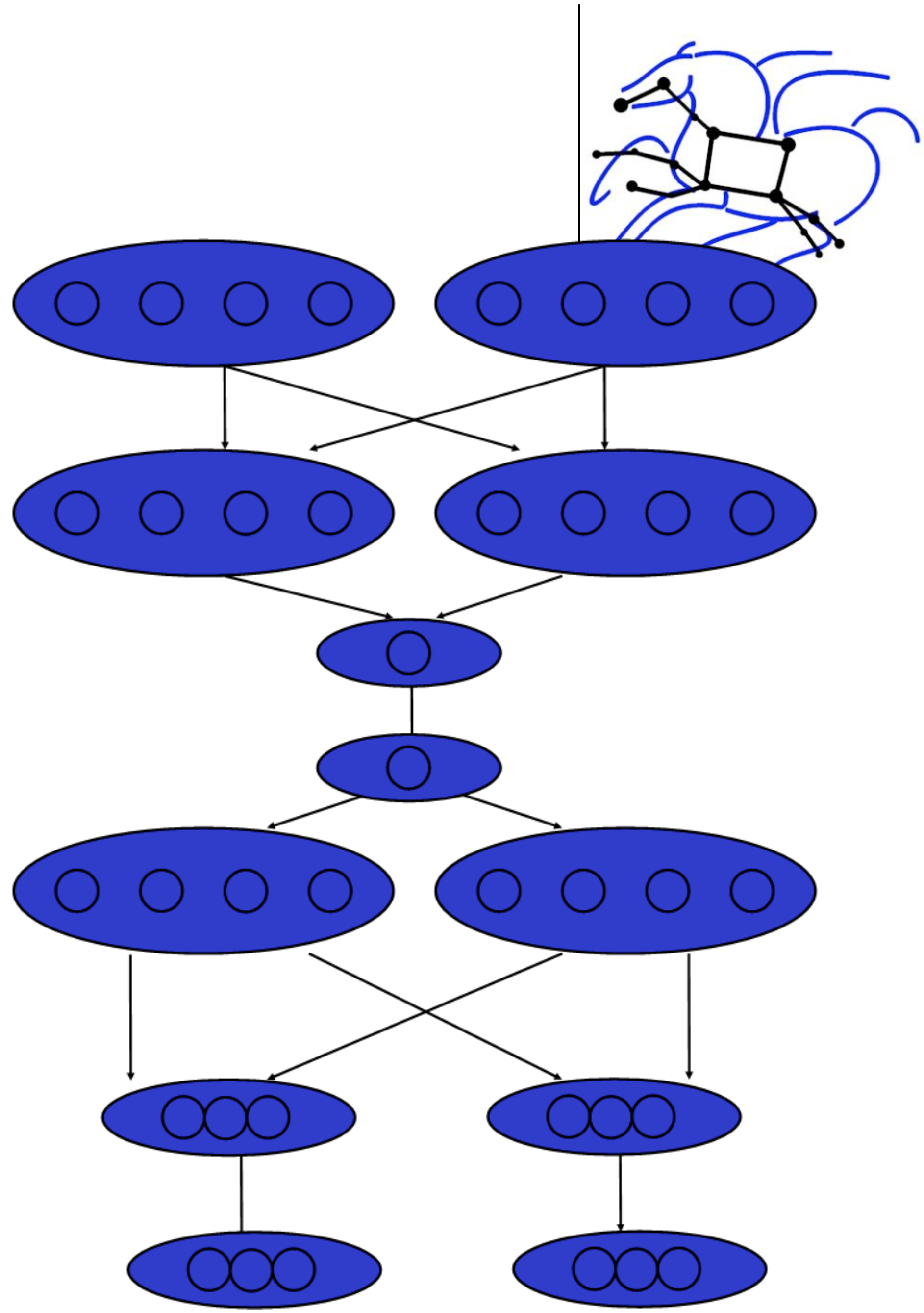


Montage workflow structure

Montage workflow of the five square degree mosaic centered at M16 region of the sky. The workflow contains 4469 jobs.

Level	Number of tasks	Runtime (in seconds)
1	892	8.2
2	2633	2
3	1	68
4	1	56
5	892	1
6	25	6
7	25	40



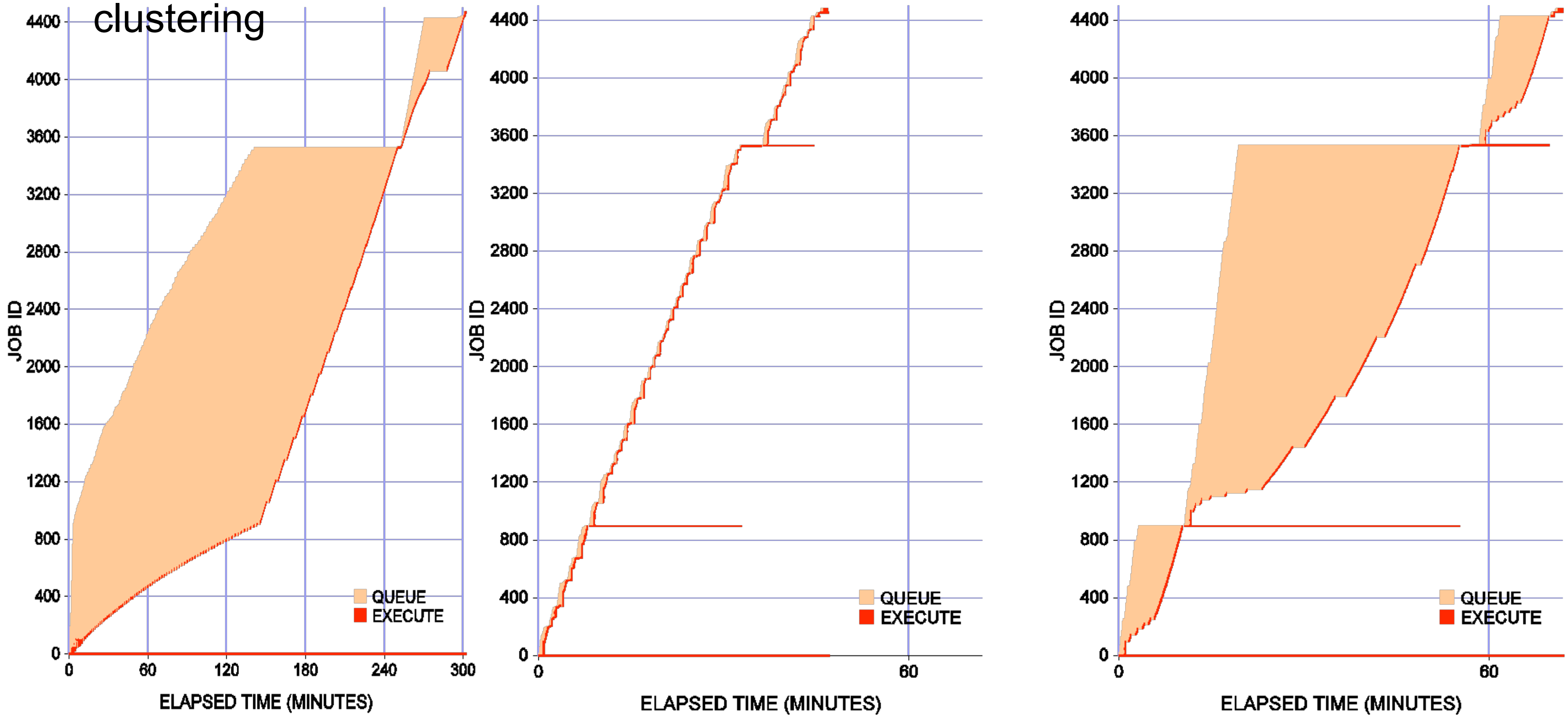


Benefits of node clustering in a Condor Pool

No clustering
clustering

1 cluster per level

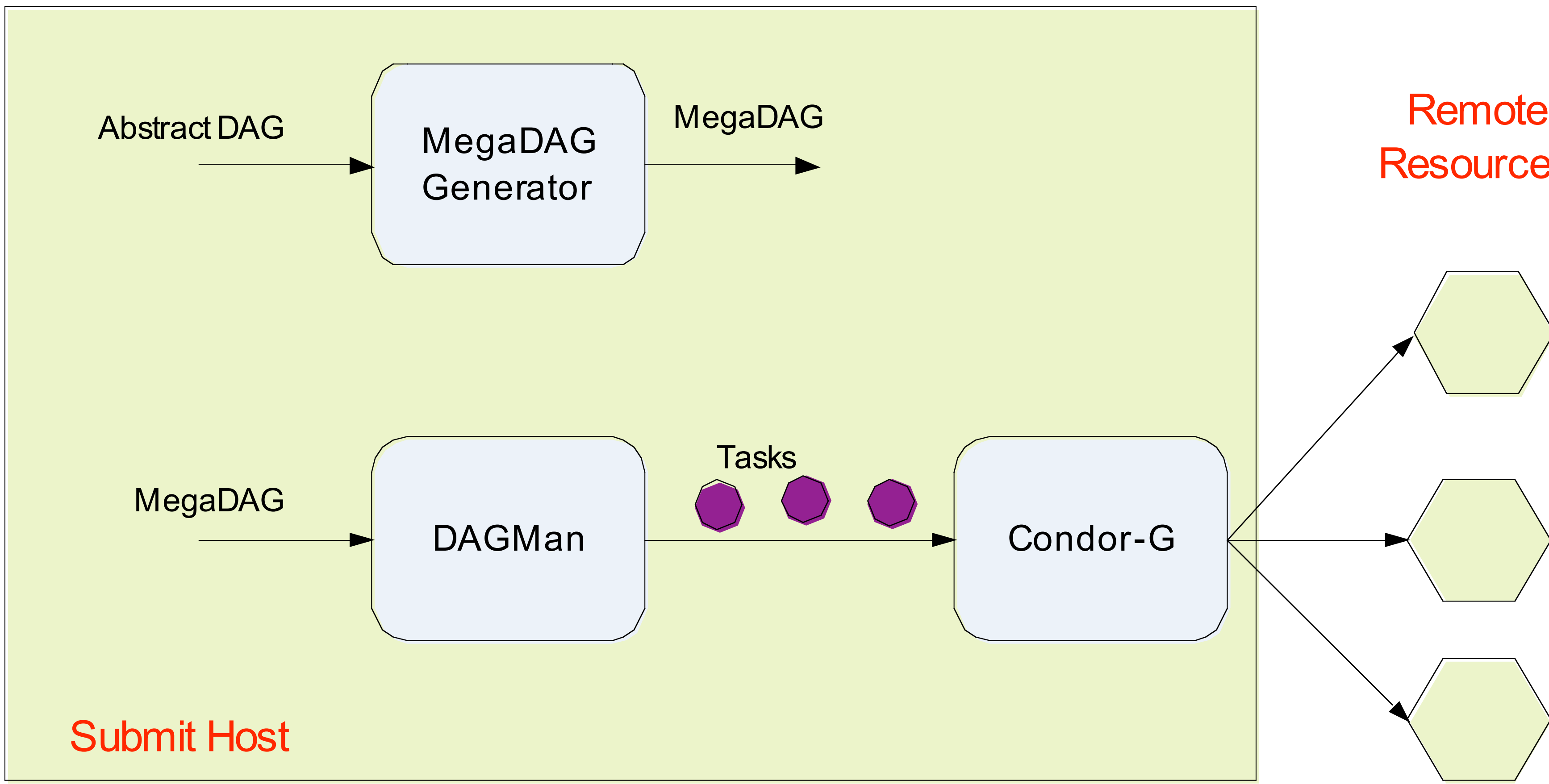
no



303 minutes

46 minutes

"Optimizing Grid-Based Workflow Execution", G. Singh, C. Kesselman, E. Deelman, JOGC


Abstract DAG
MegaDAG Generator
MegaDAG
Remote Resources
MegaDAG
DAGMan
Tasks
Condor-G
Submit Host