Workflow Management and Virtual Data

Ewa Deelman
USC Information Sciences Institute

the globus alliance
www.globus.org

Tutorial Objectives

- Provide a detailed introduction to existing services for workflow and virtual data management
- Provide descriptions and interactive demonstrations of:
  - the Chimera system for managing virtual data products
  - the Pegasus system for planning and execution in grids

Acknowledgements

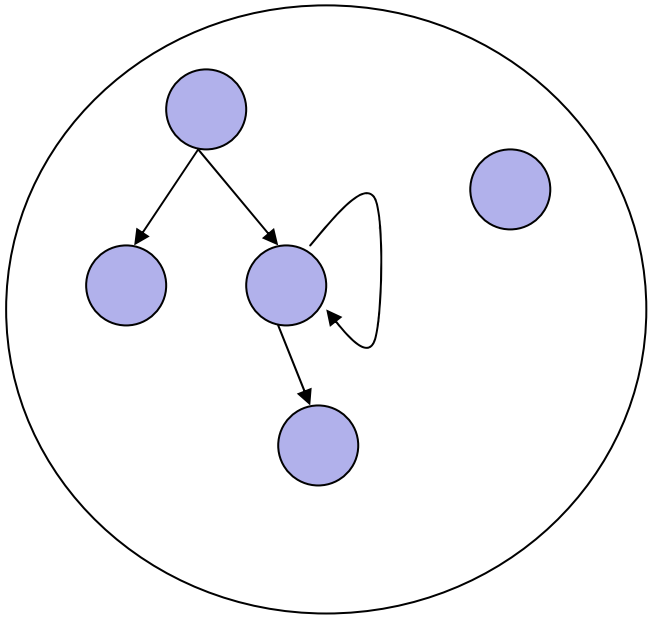
- Chimera: ANL and UofC, Ian Foster, Jens Voeckler, Mike Wilde
  - www.griphyn.org/chimera
- Pegasus: USC/ISI, Carl Kesselman, Gaurang Mehta, Gurmeet Singh, Mei-Hu Su, Karan Vahi
  - pegasus.isi.edu

Outline

- Workflows on the Grid
- The GriPhyN project
- Chimera
- Pegasus
- Research issues
- Exercises

Abstract System Representation

- A workflow is a graph
  - The vertices of the graph represent “activities”
  - The edges of the graph represent precedence between “activities”
    - > The edges are directed
  - The graph may be cyclic
  - An annotation is a set of zero or more attributes associated with an vertex, edge or subgraph of the graph

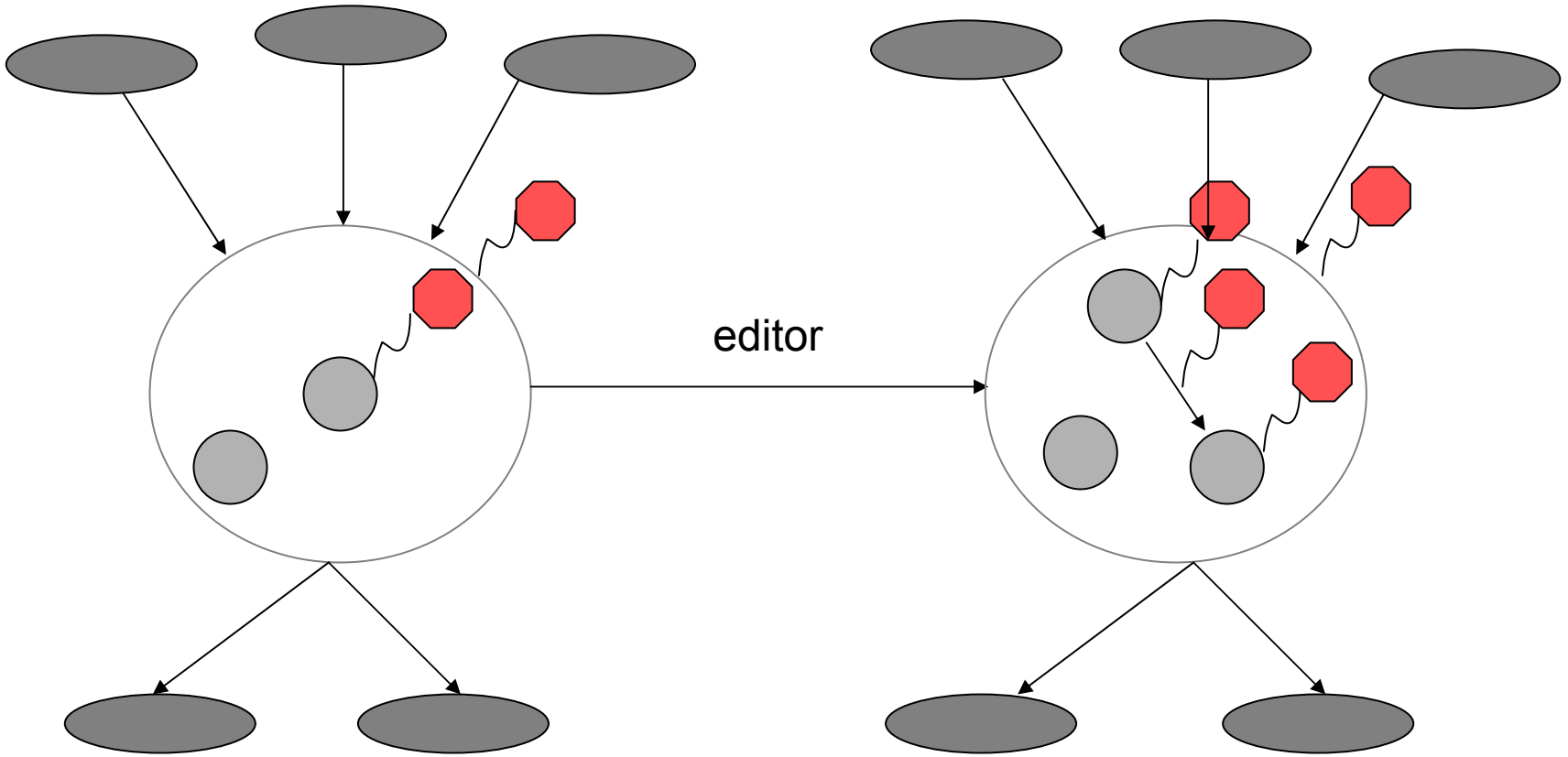


A graph

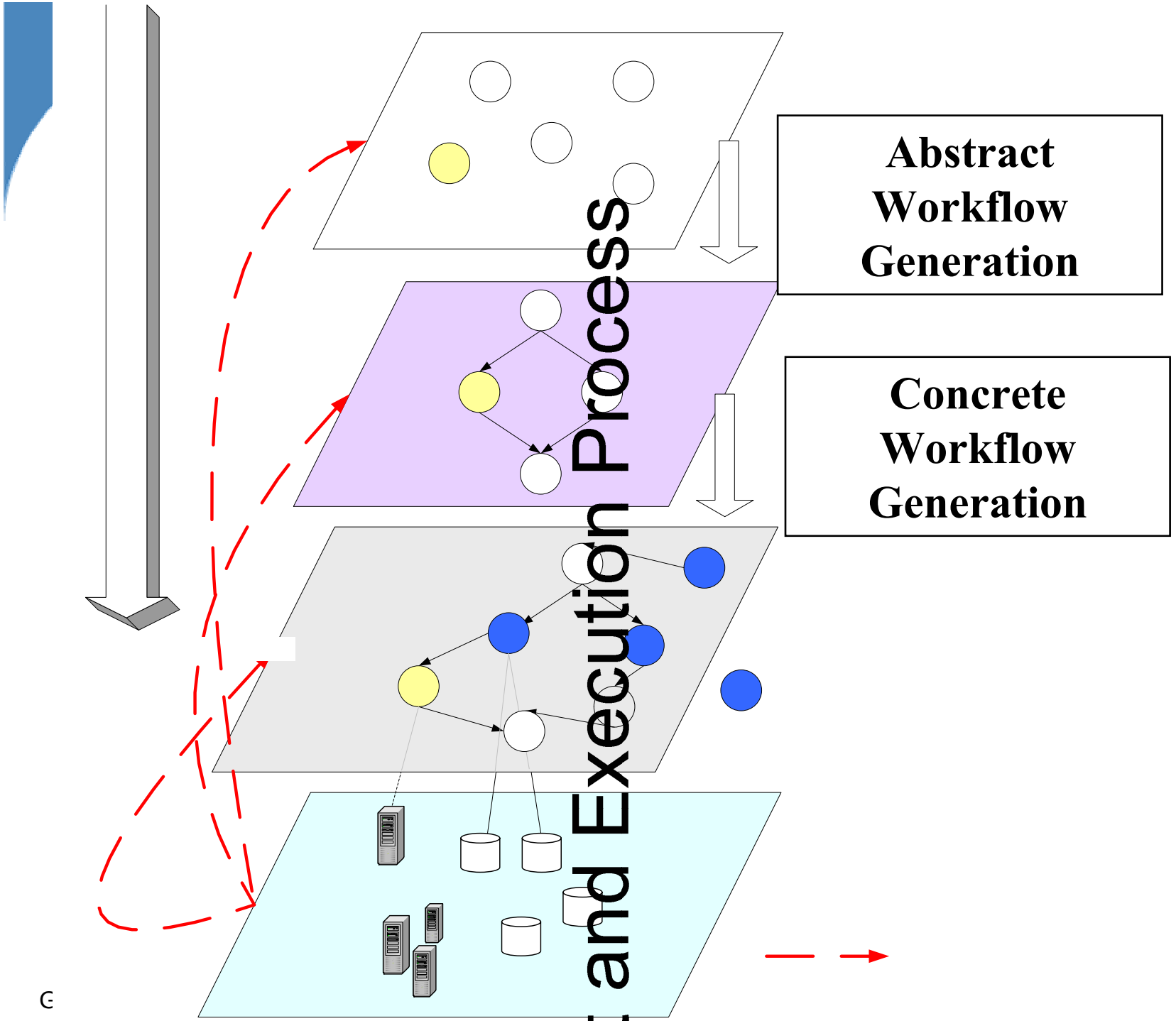
Operations on the graph

- A subgraph can be operated on by an editor
- An editor performs a transaction that maps a subgraph (s_1) onto a subgraph (s_2)
- An editor
  - May add nodes or vertices to the subgraph
  - May delete nodes or vertices within the subgraph
  - May add or modify the annotations on the subgraph or the vertices or edges in the subgraph
- After the mapping
  - the edges that were directed to s_1 are directed to s_2
  - the edges that were directed from s_1 are directed from s_2
- Two editors cannot edit two subgraphs at the same time if these subgraphs have common vertices or edges

Subgraph editing



vertex annotation Other subgraphs



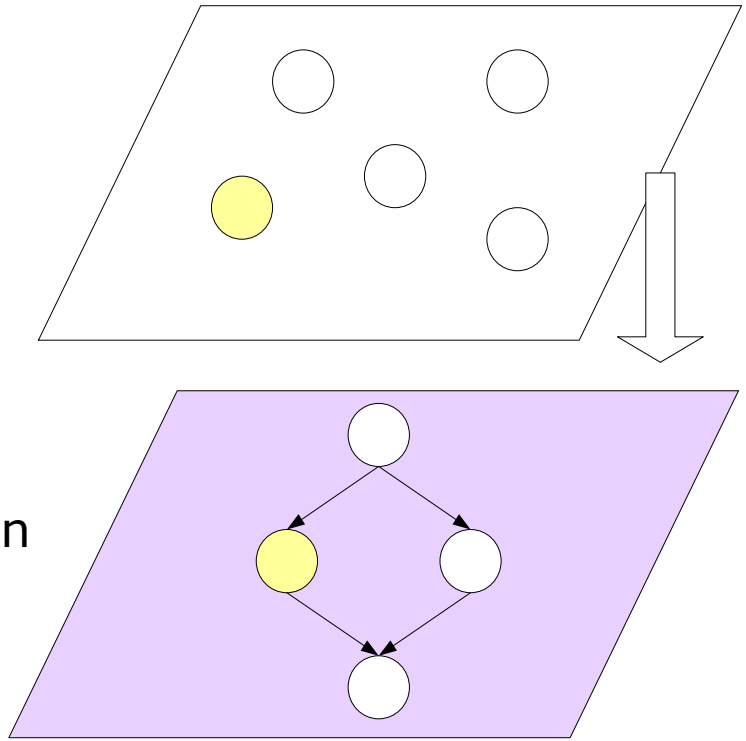
FFT

Applicati

Specify a

Generating an Abstract Workflow

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

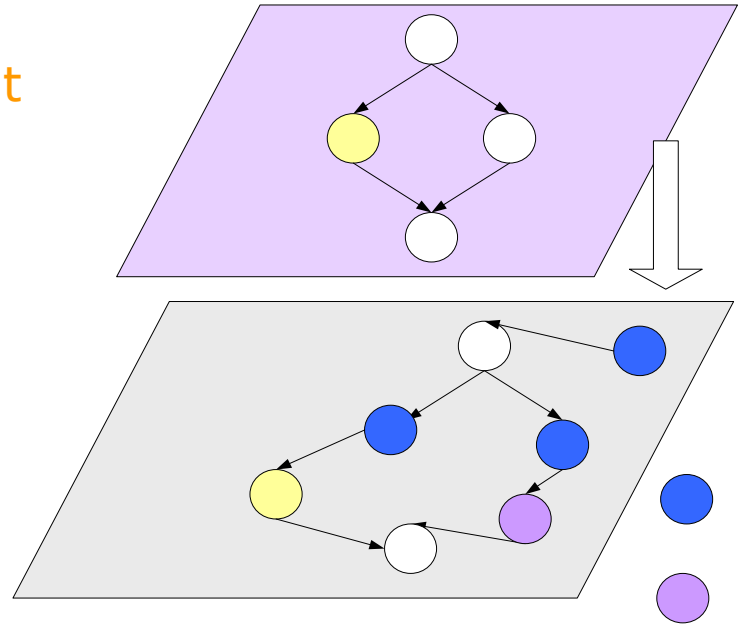


Select and configure application components to form an abstract workflow

- 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

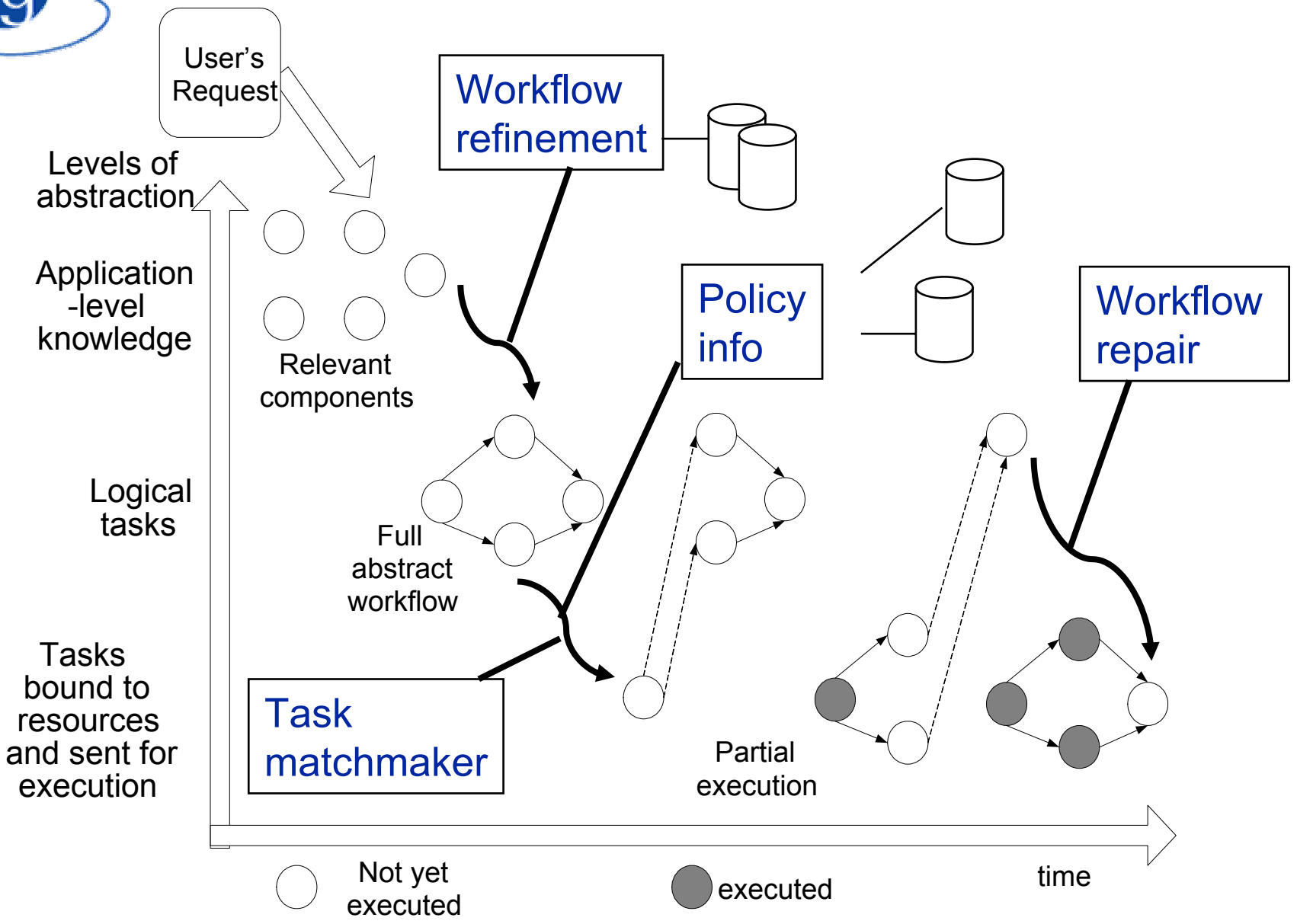
Workflow Evolution

- Workflow description
  - Metadata
  - Partial, abstract description
  - Full, abstract description
  - A concrete, executable workflow
- Workflow refinement
  - Take a description and produce an executable workflow
- Workflow execution

Workflow Refinement

- The workflow can undergo an arbitrarily complex set of refinements
- A refiner can modify part of the workflow or the entire workflow
- A refiner uses a set of Grid information services and catalogs to perform the refinement (metadata catalog, virtual data catalog, replica location services, monitoring and discovery services, etc.)

Workflow Refinement and execution



Outline

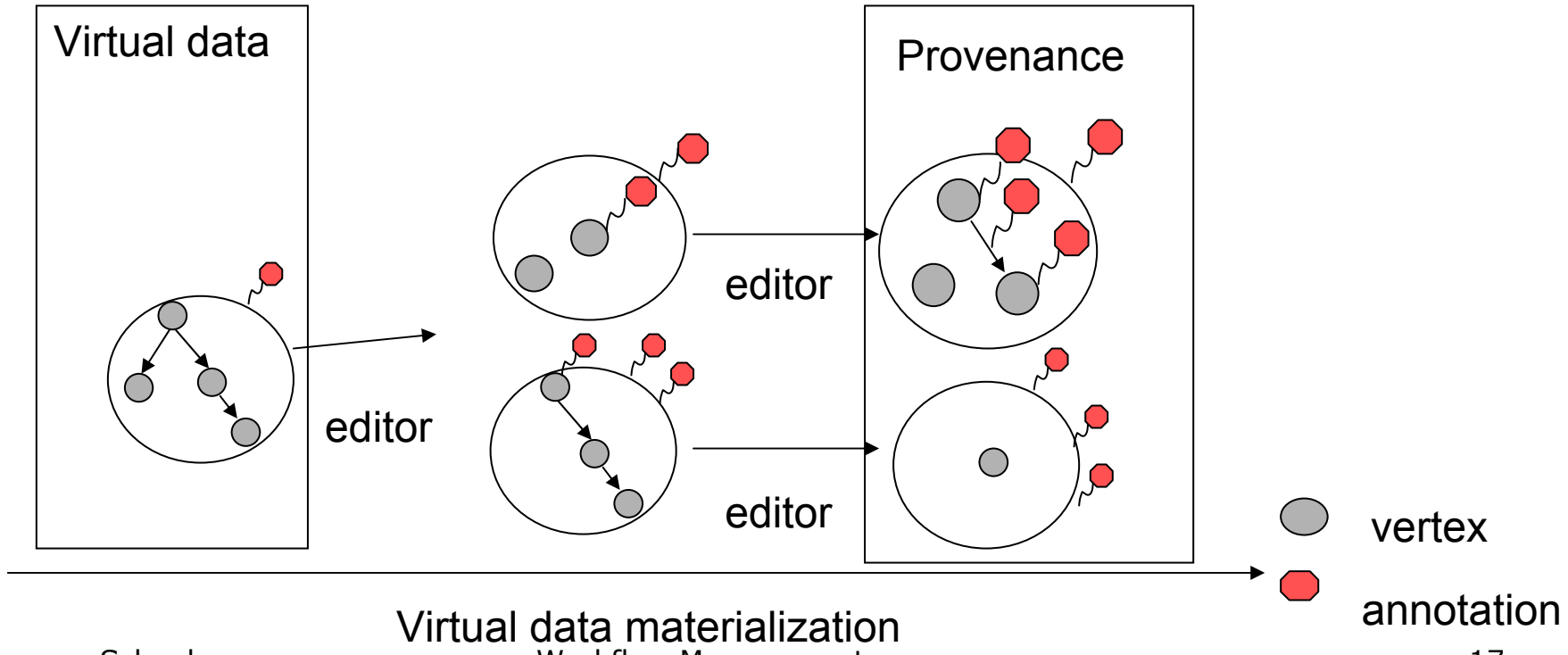
- Workflows on the Grid
- The GriPhyN project
- Chimera
- Pegasus
- Exercises

Ongoing Workflow Management Work

- Part of the NSF-funded GriPhyN project
- Supports the concept of Virtual Data, where data is materialized on demand
- Data can exist on some data resource and be directly accessible
- Data can exist only in a form of a recipe
- The GriPhyN Virtual Data System can seamlessly deliver the data to the user or application regardless of the form in which the data exists
- GriPhyN targets applications in high-energy physics, gravitational-wave physics and astronomy

Relationship between virtual data, and provenance

- Virtual data can be described by a subgraph, that needs to undergo an editing process to obtain a subgraph in the state that is "done"
- The recoding of the editing process is provenance



Workflow management in GriPhyN

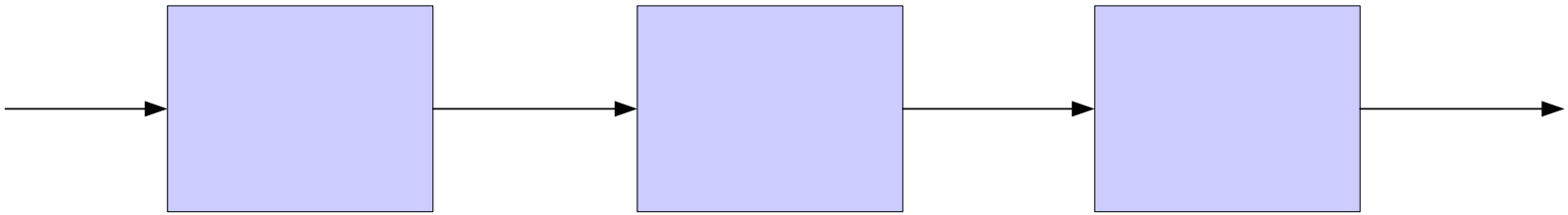
- Workflow Generation: how do you describe the workflow (at various levels of abstraction)? (Chimera)
- Workflow Mapping/Refinement: how do you map an abstract workflow representation to an executable form? (Pegasus)
- Workflow Execution: how to you reliably execute the workflow? (Condor's DAGMan)

Terms

- Abstract Workflow (DAX)
  - Expressed in terms of logical entities
  - Specifies all logical files required to generate the desired data product from scratch
  - Dependencies between the jobs
  - Analogous to build style dag
- Concrete Workflow
  - Expressed in terms of physical entities
  - Specifies the location of the data and executables
  - Analogous to a make style dag

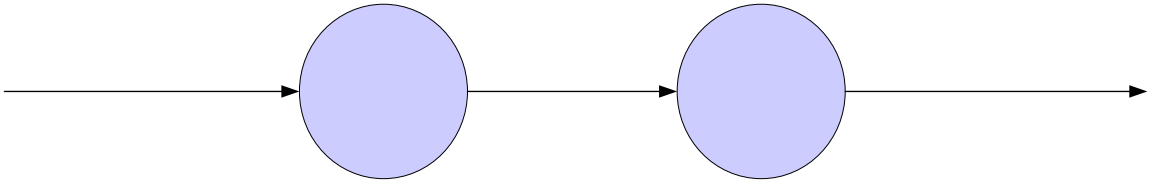
Executable Workflow Construction

- Chimera builds an abstract workflow based on VDL descriptions
- Pegasus takes the abstract workflow and produces and executable workflow for the Grid
- Condor's DAGMan executes the workflow

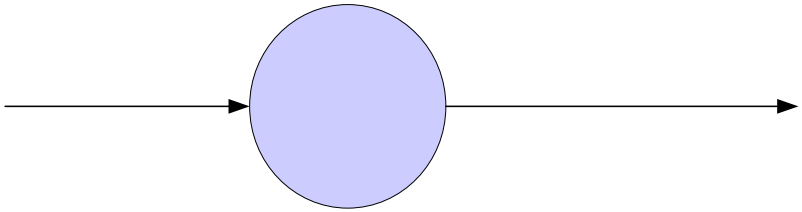


Example Workflow Reduction

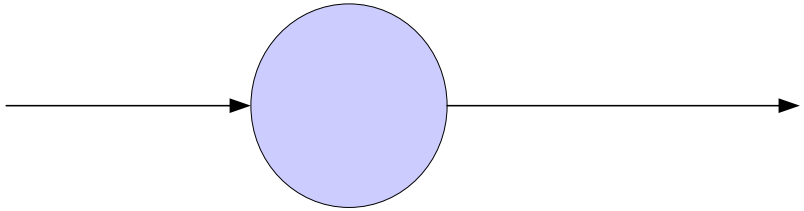
- Original abstract workflow



- 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

the globus alliance
www.globus.org

Application Workflow Characteristics

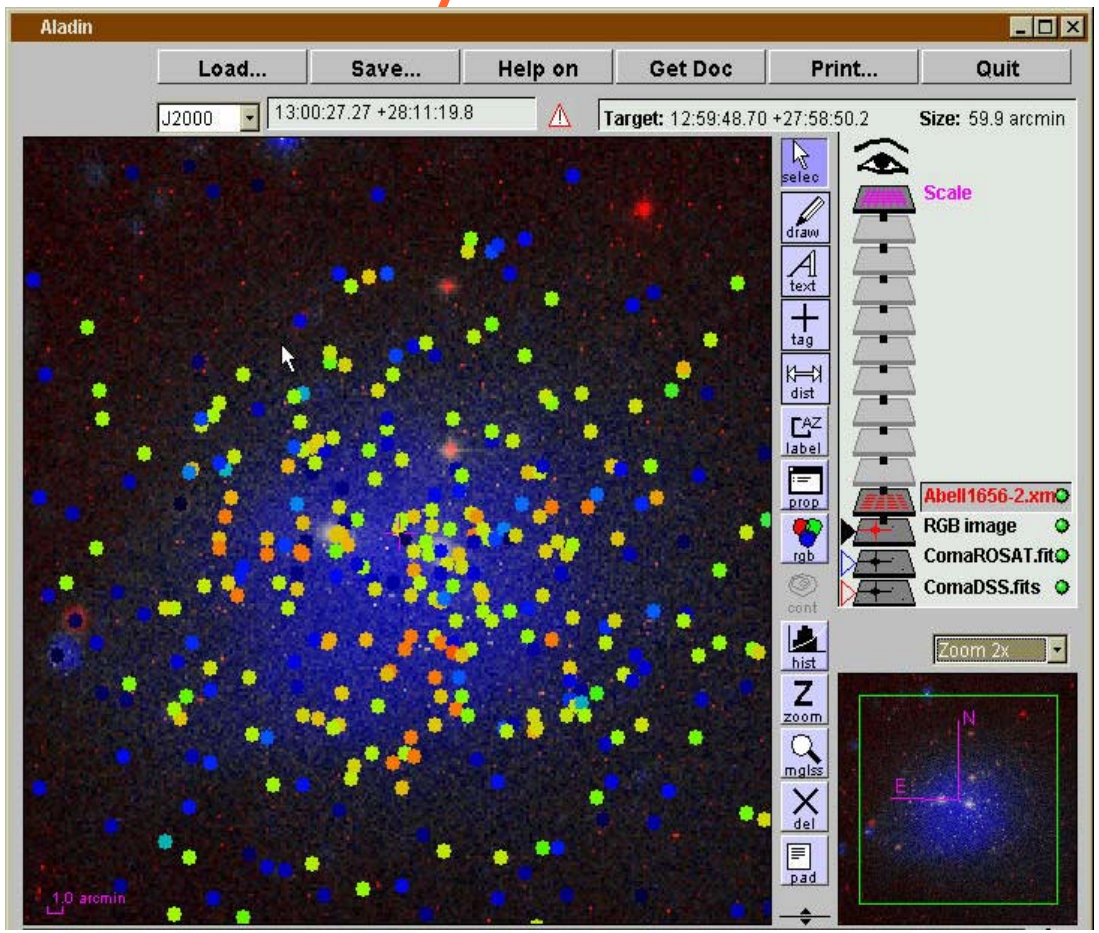
Experiment	#workflows per analysis	# of jobs in workflow	Data Size per job	Compute Time per job
LHC	O(100K)	7	~300MB	~12CPU hours
LIGO	O(1K)	100-400	~1MB	~2min
SDSS	O(20K)	10	~1MB	~1-5 min

Number of resources:

currently several condor pools and clusters with 100s of nodes

Astronomy

- Galaxy Morphology (National Virtual Observatory)
  - Investigates the dynamical state of galaxy clusters
  - Explores galaxy evolution inside the context of large-scale structure.
  - Uses galaxy morphologies as a probe of the star formation and stellar distribution history of the galaxies inside the clusters.
  - Data intensive computations involving hundreds of galaxies in a cluster



The x-ray emission is shown in blue, and the optical mission is in red. The colored dots are located at the positions of the galaxies within the cluster; the dot color represents the value of the asymmetry index. Blue dots represent the most asymmetric galaxies and are scattered throughout the image, while orange are the most symmetric, indicative of elliptical galaxies, are concentrated more toward the center.

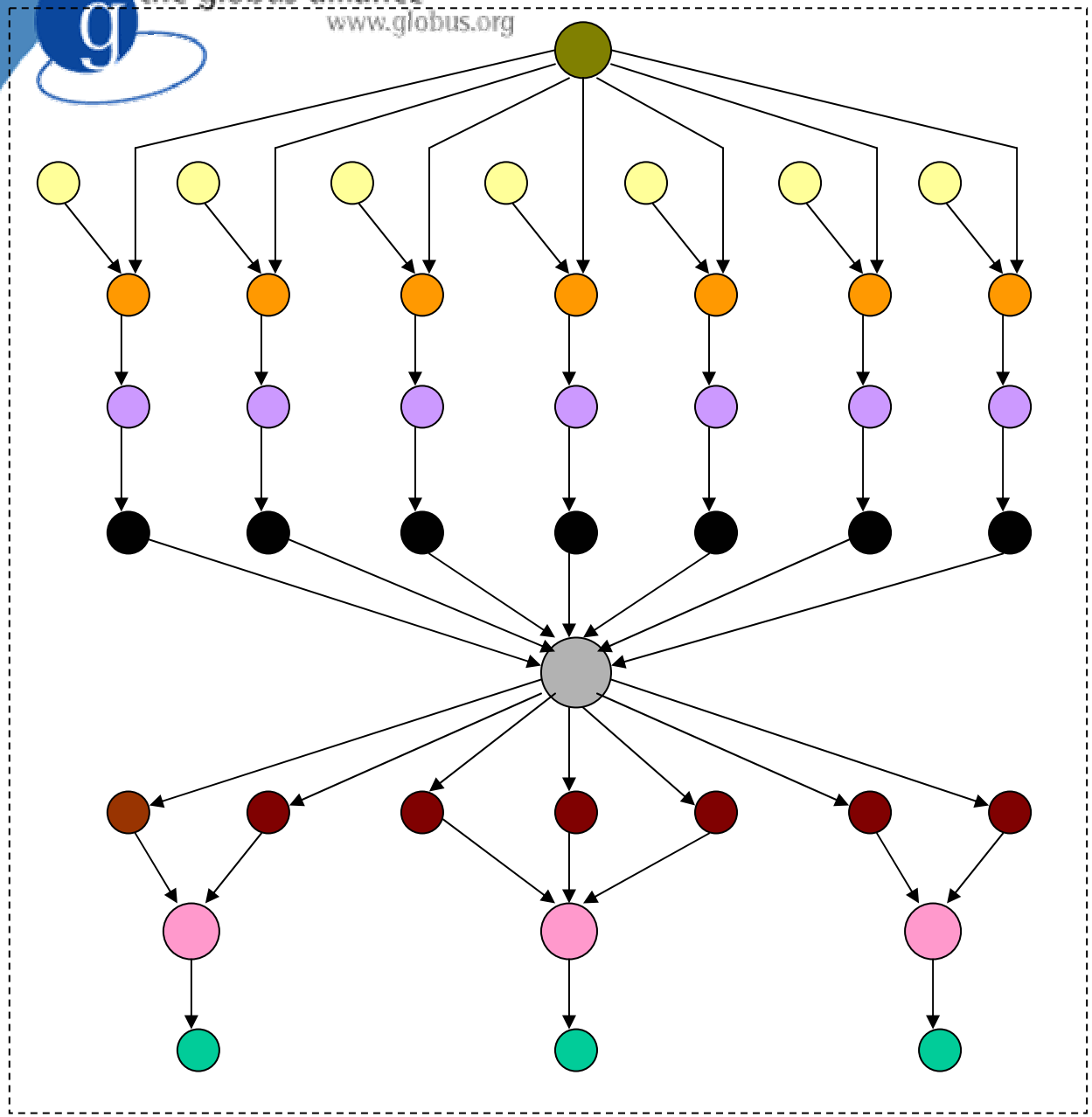
People involved: Gurmeet Singh, Mei-Hui Su, many others

Astronomy

- Sloan Digital Sky Survey (GriPhyN project)
  - finding clusters of galaxies from the Sloan Digital Sky Survey database of galaxies.
  - Lead by Jim Annis (Fermi), Mike Wilde (ANL)

- Montage (NASA and NVO) (Bruce Berriman, John Good, Joe Jacob, Gurmeet Singh, Mei-Hui Su)
  - 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.

Montage Workflow



Transfer the template header

Transfer the image file

Re-projection of images.

Calculating the difference

Fit to a common plane

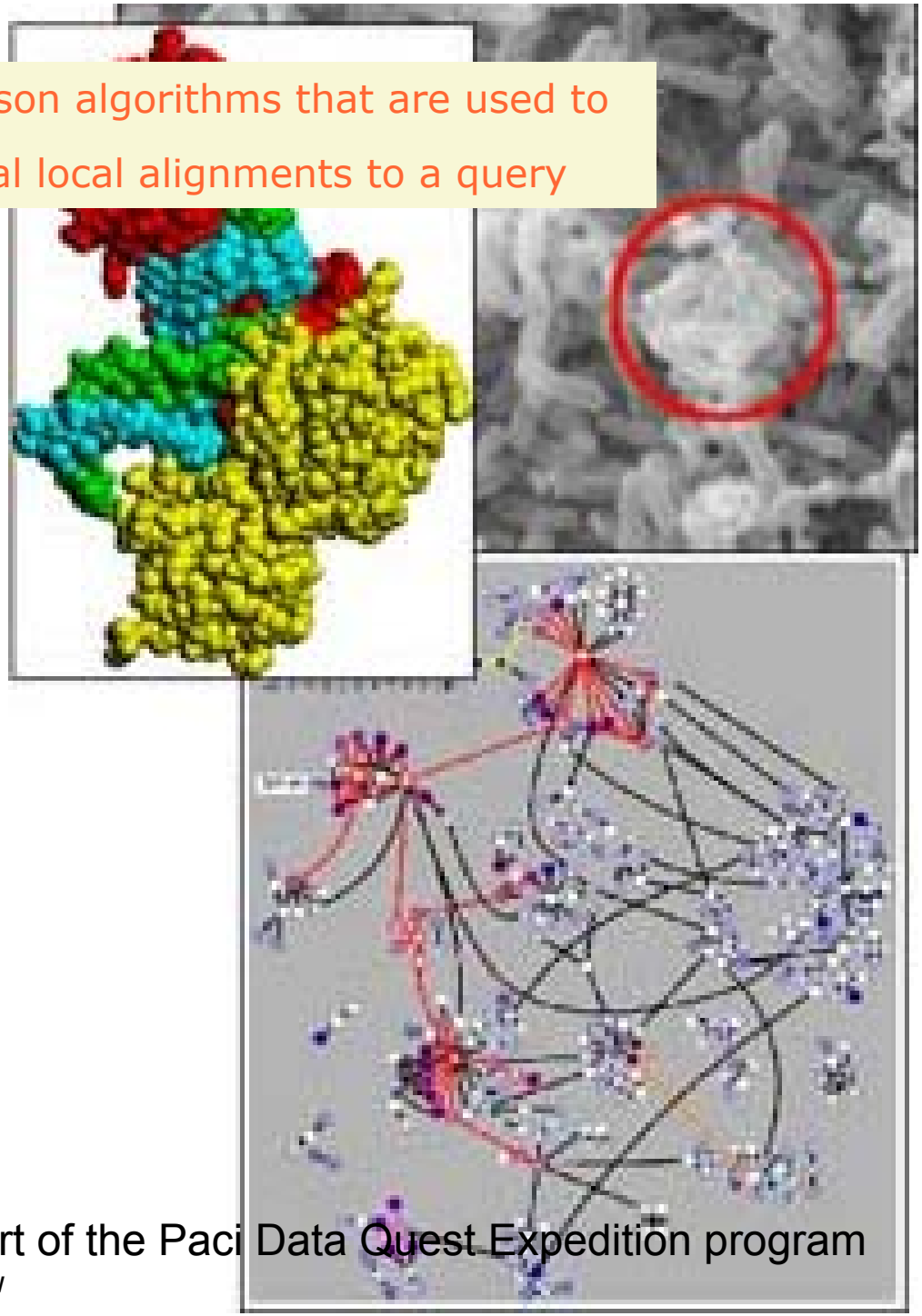
Background modeling

Background correction

Adding the images to get the final mosaic

Register the mosaic in RLS

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



2 major runs were performed using Chimera and Pegasus:

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

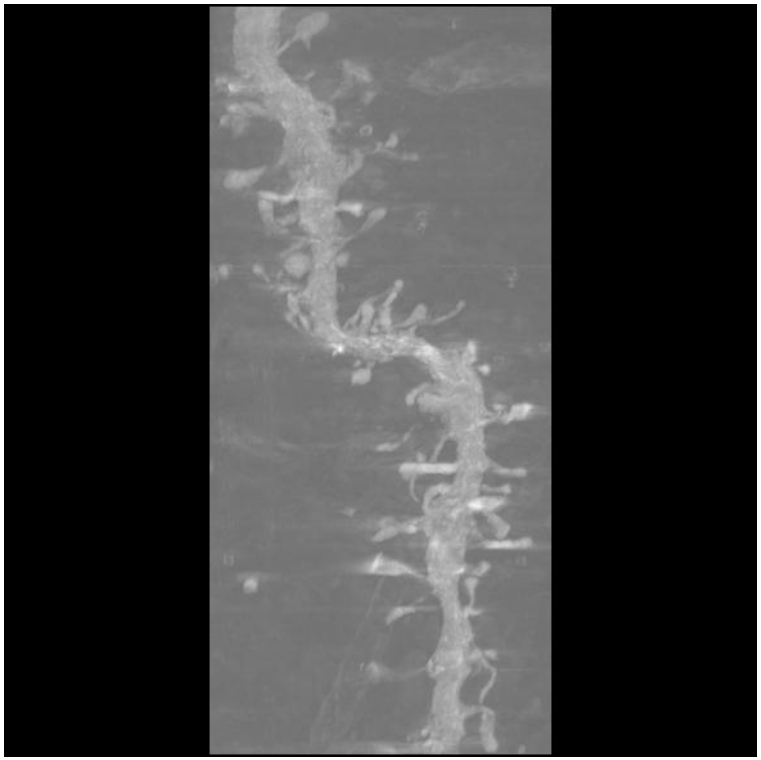
Speedup of 5-20 times were achieved because the compute nodes we used efficiently by keeping the submission of the jobs to the compute cluster constant.

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

Biology Applications (cont'd)

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

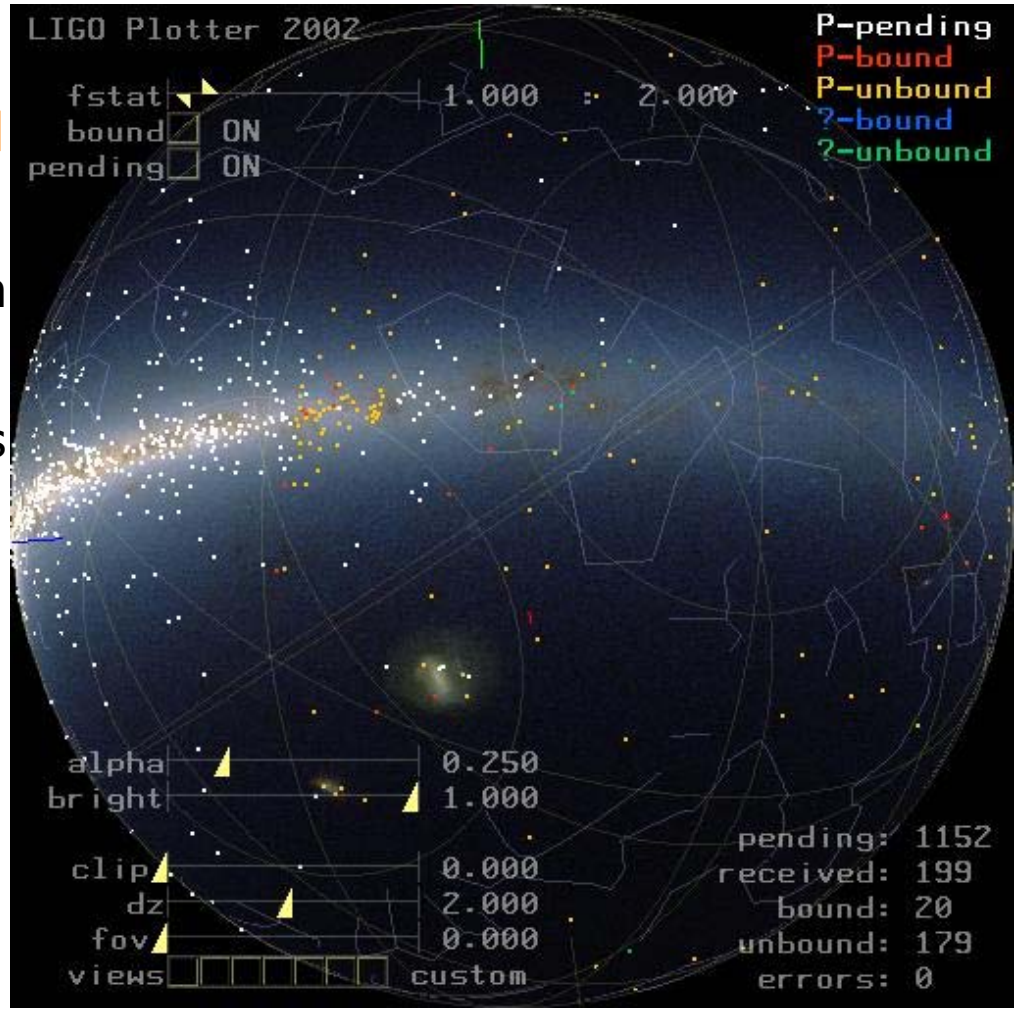
Work performed by Mei Hui-Su with Mark Ellisman, Steve Peltier, Abel Lin, Thomas Molina (SDSC)

Physics (GriPhyN Project)

- High-energy physics
  - CMS—collaboration with Rick Cavannaugh, UFL
    - > Processed simulated events
    - > Cluster of 25 dual-processor Pentium machines.
    - > Computation: 7 days, 678 jobs with 250 events each
    - > Produced ~ 200GB of simulated data.
  - Atlas
    - > Uses GriPhyN technologies for production Rob Gardner
- Gravitational-wave science (collaboration with Bruce Allen A. Lazzarini and S. Koranda)

LIGO's pulsar search at SC 2002

- The pulsar search conducted at SC 2002
  - Used LIGO's data collected during the first scientific run of the instrument
  - Targeted a set of 1000 locations of known pulsar as well as random locations in the sky
  - Results of the analysis were be published via LDAS (LIGO Data Analysis System) to the LIGO Scientific Collaboration
  - performed using LDAS and compute and storage resources at Caltech, University of Southern California, University of Wisconsin Milwaukee.



ISI people involved: Gaurang Mehta, Sonal Patil, Srividya Rao, Gurmeet Singh, Karan Vahi
Visualization by Marcus Thiebaux

the globus alliance
www.globus.org

Outline

- Workflows on the Grid
- The GriPhyN project
- Chimera
- Pegasus
- Research issues
- Exercises

the globus alliance
www.globus.org

Chimera Virtual Data System
Outline

- Virtual data concept and vision
- VDL – the Virtual Data Language
- Simple virtual data examples
- Virtual data applications in High Energy Physics and Astronomy

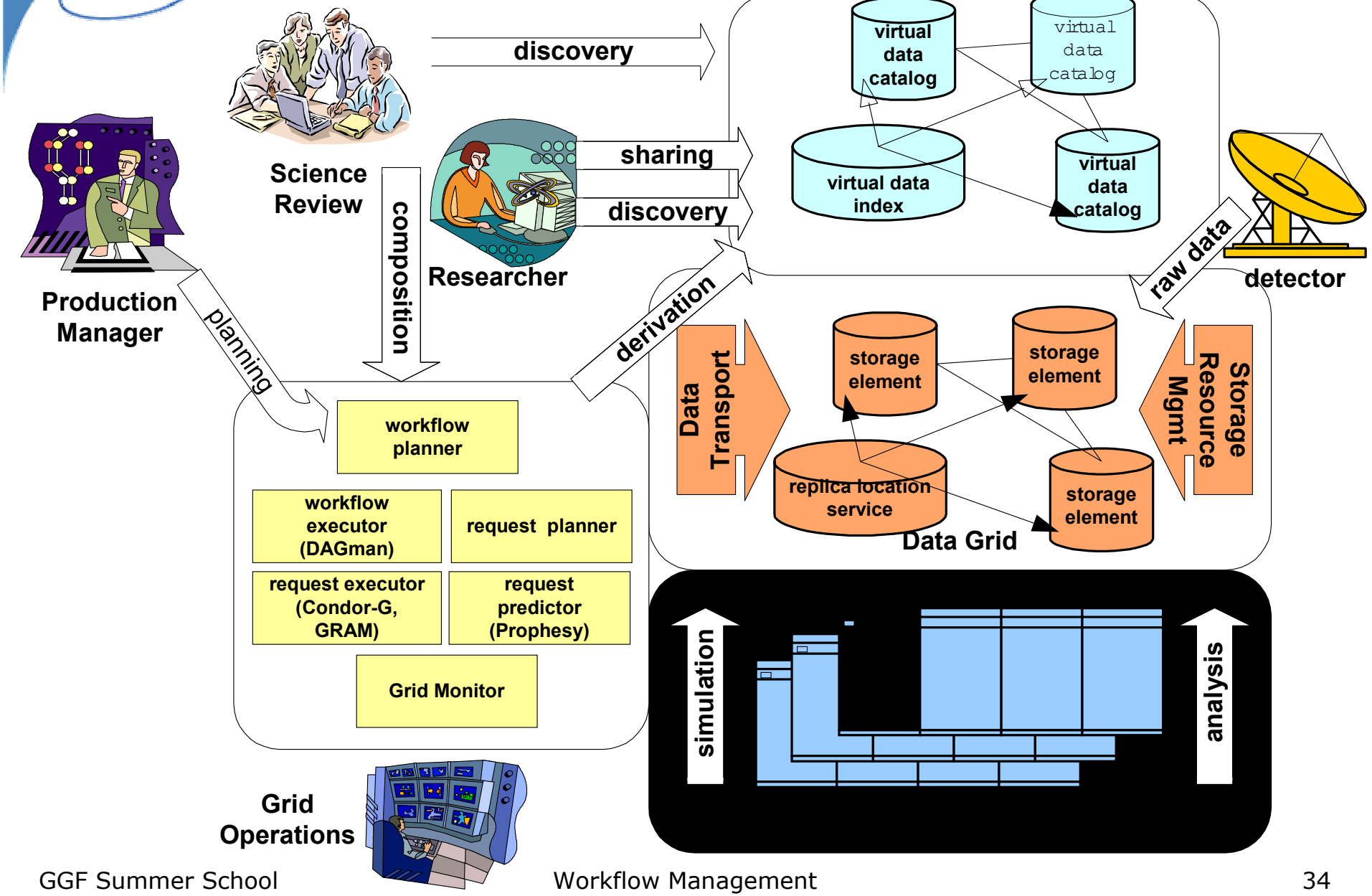
The Virtual Data Concept

Enhance scientific productivity through:

- Discovery and application of datasets and programs at petabyte scale
- Enabling use of a worldwide data grid as a scientific workstation

Virtual Data enables this approach by creating datasets from workflow "recipes" and recording their provenance.

Virtual Data Vision



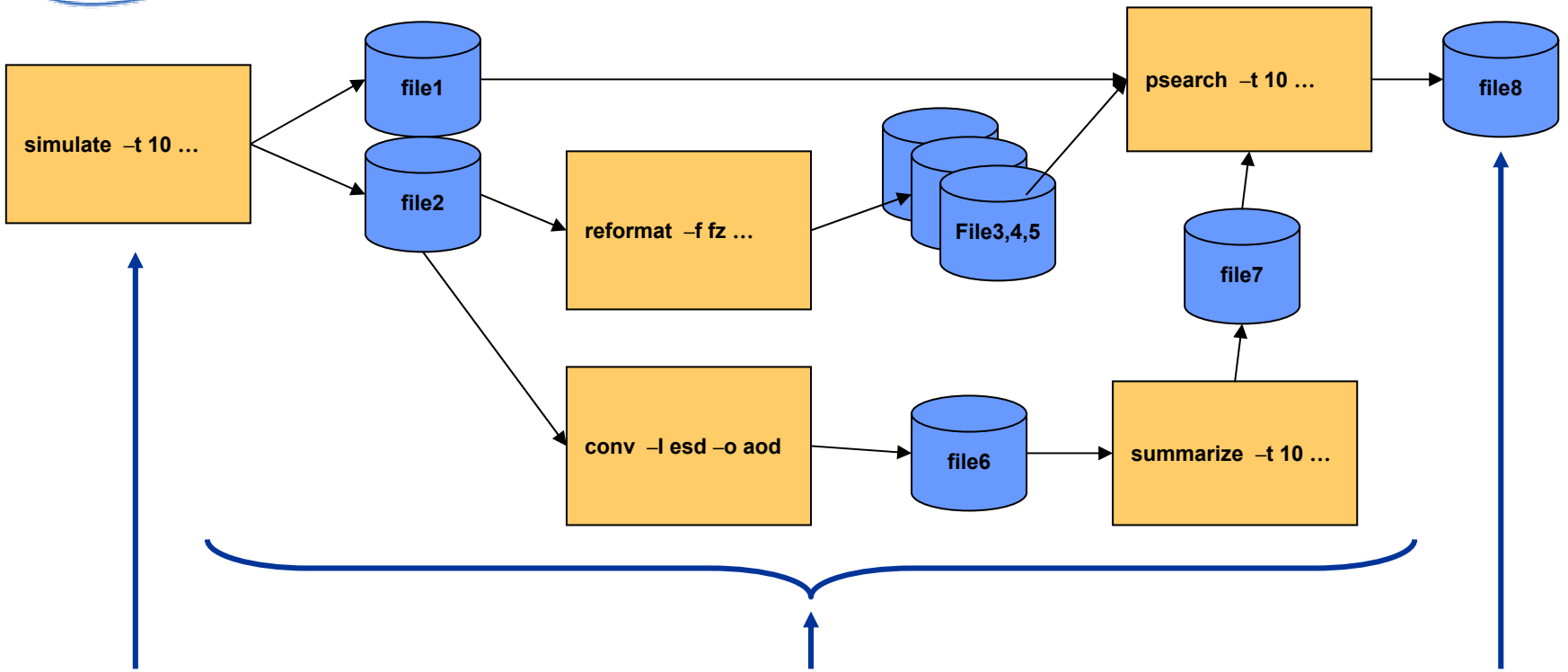
the globus alliance
www.globus.org

Virtual Data System Capabilities

Producing data from transformations with uniform, precise data interface descriptions enables…

- *Discovery*: finding and understanding datasets and transformations
- *Workflow*: structured paradigm for organizing, locating, specifying, & producing scientific datasets
  - Forming new workflow
  - Building new workflow from existing patterns
  - Managing change
- *Planning*: automated to make the Grid transparent
- *Audit*: explanation and validation via provenance

Virtual Data Scenario



Update workflow following changes

Manage workflow;
Explain provenance, e.g. for file8:

psearch –t 10 –i file3 file4 file5 –o file8
summarize –t 10 –i file6 –o file7
reformat –f fz –i file2 –o file3 file4 file5
conv –I esd –o aod –i file 2 –o file6
simulate –t 10 –o file1 file2

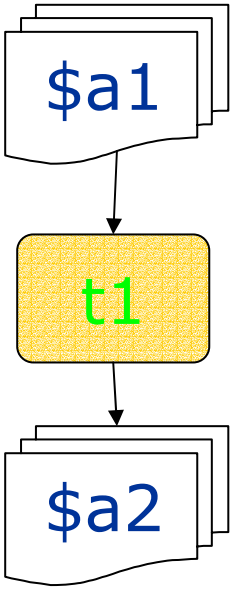
On-demand data generation

VDL: Virtual Data Language Describes Data Transformations

- Transformation
  - Abstract template of program invocation
  - Similar to "function definition"
- Derivation
  - “Function call” to a Transformation
  - Store past and future:
    - A record of how data products were generated
    - A recipe of how data products can be generated
- Invocation
  - Record of a Derivation execution

Example Transformation

```
TR t1( out a2, in a1, none pa = "500", none
   env = "100000" ) {
  argument = "-p "${pa};
  argument = "-f "${a1};
  argument = "-x –y";
  argument stdout = ${a2};
  profile env.MAXMEM = ${env};
}
```

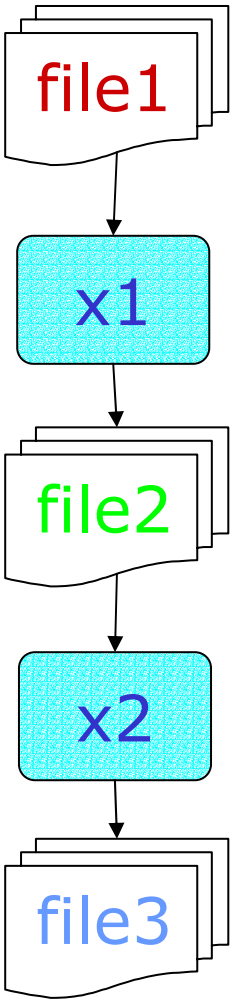


Example Derivations

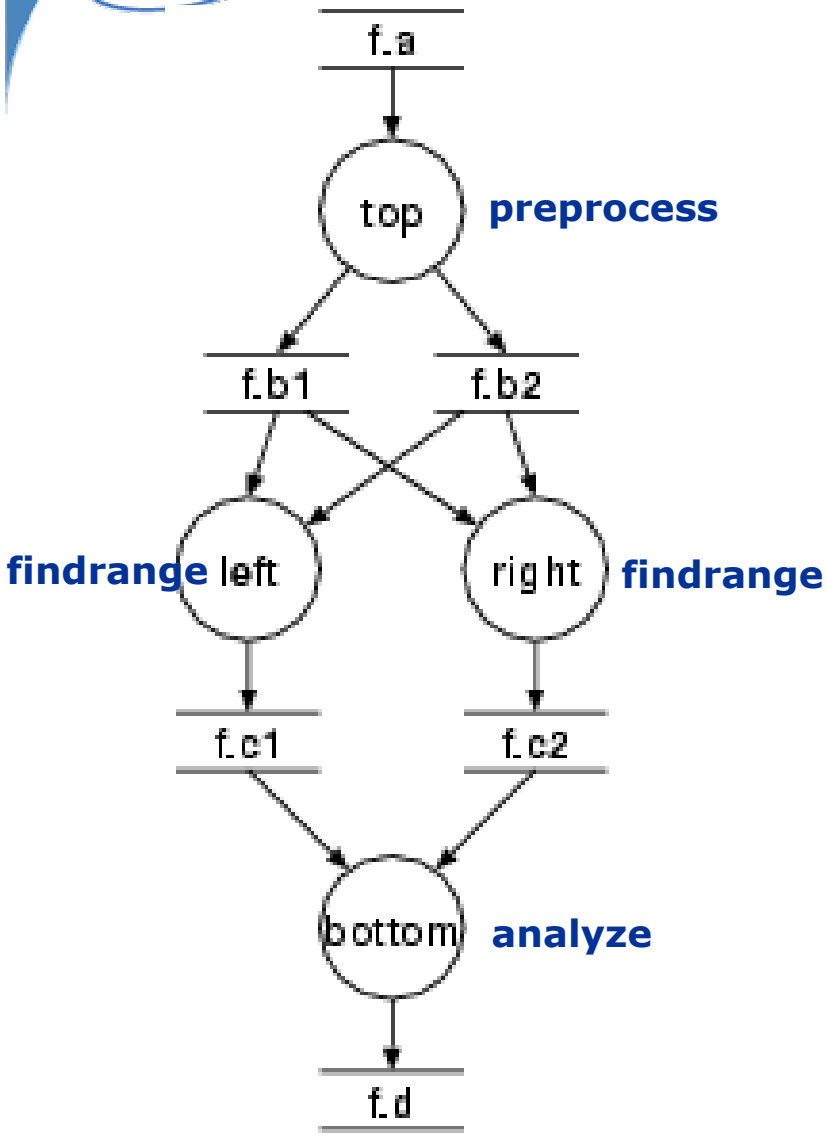
```
DV d1->t1 (
  env="20000", pa="600",
  a2=@{out:run1.exp15.T1932.summary},
  a1=@{in:run1.exp15.T1932.raw},
);

DV d2->t1 (
  a1=@{in:run1.exp16.T1918.raw},
  a2=@{out.run1.exp16.T1918.summary}
);
```

Workflow from File Dependencies

```
TR tr1(in a1, out a2) {
    argument stdin = ${a1};
    argument stdout = ${a2}; }

TR tr2(in a1, out a2) {
    argument stdin = ${a1};
    argument stdout = ${a2}; }

DV x1->tr1(a1=@{in:file1}, a2=@{out:file2});
DV x2->tr2(a1=@{in:file2}, a2=@{out:file3});
```



Example Workflow



- Complex structure
  - Fan-in
  - Fan-out
  - "left" and "right" can run in parallel
- Uses input file
  - Register with RC
- Complex file dependencies
  - Glues workflow

Workflow step "preprocess"

- TR preprocess turns f.a into f.b1 and f.b2

```
TR preprocess( output b[], input a ) {
  argument = "-a top";
  argument = " –i "${input:a};
  argument = " –o " ${output:b};
}
```

- Makes use of the "list" feature of VDL
  - Generates 0..N output files.
  - Number file files depend on the caller.

Workflow step "findrange"

- Turns two inputs into one output

```
TR findrange( output b, input a1, input a2,
  none name="findrange", none p="0.0" ) {
  argument = "-a "${name};
  argument = " –i " ${a1} " " ${a2};
  argument = " –o " ${b};
  argument = " –p " ${p};
}
```

- Uses the default argument feature

the globus alliance
www.globus.org

Can also use list[] parameters

```
TR findrange( output b, input a[],
  none name="findrange", none p="0.0" ) {
  argument = "-a "${name};
  argument = " –i " ${" "|a};
  argument = " –o " ${b};
  argument = " –p " ${p};
}
```

Workflow step "analyze"

- Combines intermediary results

```
TR analyze( output b, input a[] ) {
  argument = "-a bottom";
  argument = " –i " ${a};
  argument = " –o " ${b};
}
```

Complete VDL workflow

- Generate appropriate derivations

```
DV top->preprocess( b=[ @{out:"f.b1"}, @{
  out:"f.b2"} ], a=@{in:"f.a"} );
DV left->findrange( b=@{out:"f.c1"},
  a2=@{in:"f.b2"}, a1=@{in:"f.b1"},
  name="left", p="0.5" );
DV right->findrange( b=@{out:"f.c2"},
  a2=@{in:"f.b2"}, a1=@{in:"f.b1"},
  name="right" );
DV bottom->analyze( b=@{out:"f.d"}, a=[
  @{in:"f.c1"}, @{in:"f.c2"} );
```

Compound Transformations

- Using compound TR
  - Permits composition of complex TRs from basic ones
  - Calls are independent
    - unless linked through LFN
  - A Call is effectively an anonymous derivation
    - Late instantiation at workflow generation time
  - Permits bundling of repetitive workflows
  - Model: Function calls nested within a function definition

Compound Transformations (cont)

- TR diamond bundles black-diamonds

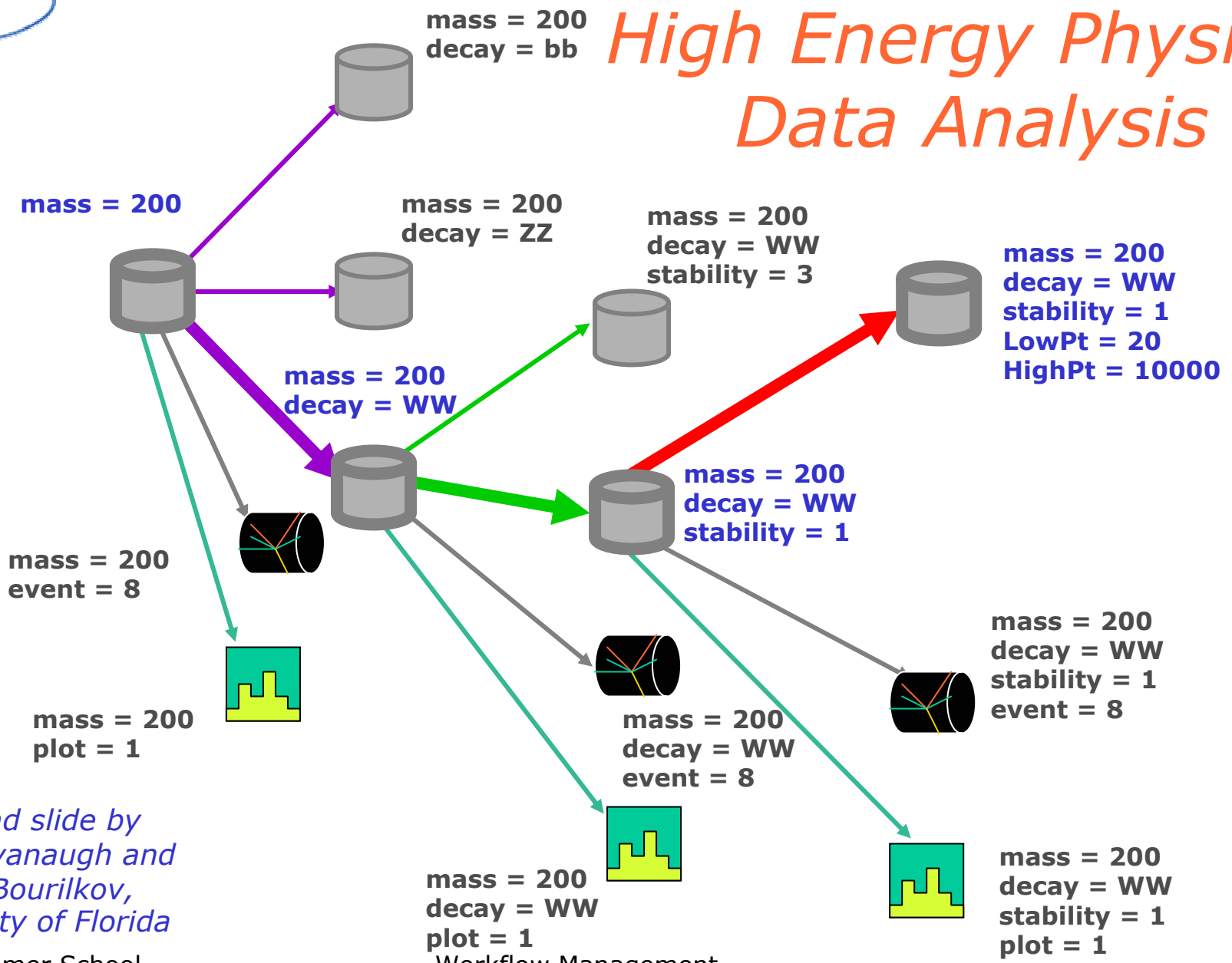
```
TR diamond( out fd, io fc1, io fc2, io fb1, io fb2, in fa, p1,
  p2 ) {
 call preprocess( a=${fa}, b=[ ${out:fb1}, ${out:fb2} ]
  );
 call findrange( a1=${in:fb1}, a2=${in:fb2},
  name="LEFT", p=${p1}, b=${out:fc1} );
 call findrange( a1=${in:fb1}, a2=${in:fb2},
  name="RIGHT", p=${p2}, b=${out:fc2} );
 call analyze( a=[ ${in:fc1}, ${in:fc2} ], b=${fd} );
}
```

Compound Transformations (cont)

- Multiple DVs allow easy generator scripts:

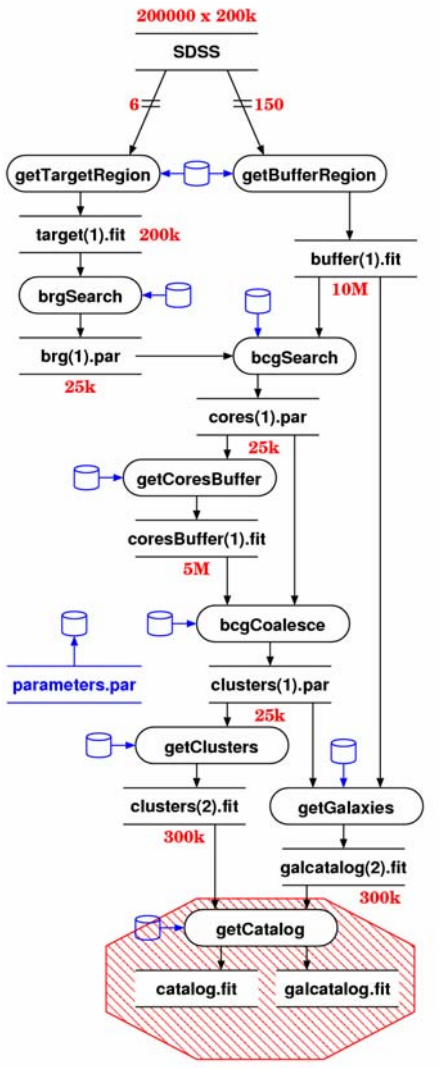
```
DV d1->diamond( fd=@{out:"f.00005"},
   fc1=@{io:"f.00004"}, fc2=@{io:"f.00003"},
   fb1=@{io:"f.00002"}, fb2=@{io:"f.00001"},
   fa=@{io:"f.00000"}, p2="100", p1="0" );
DV d2->diamond( fd=@{out:"f.0000B"},
   fc1=@{io:"f.0000A"}, fc2=@{io:"f.00009"},
   fb1=@{io:"f.00008"}, fb2=@{io:"f.00007"},
   fa=@{io:"f.00006"}, p2="141.4213562373l", p1="0" );
...
DV d70->diamond( fd=@{out:"f.001A3"},
   fc1=@{io:"f.001A2"}, fc2=@{io:"f.001A1"},
   fb1=@{io:"f.001A0"}, fb2=@{io:"f.0019F"},
   fa=@{io:"f.0019E"}, p2="800", p1="18" );
```

Virtual Data Application: *High Energy Physics Data Analysis*



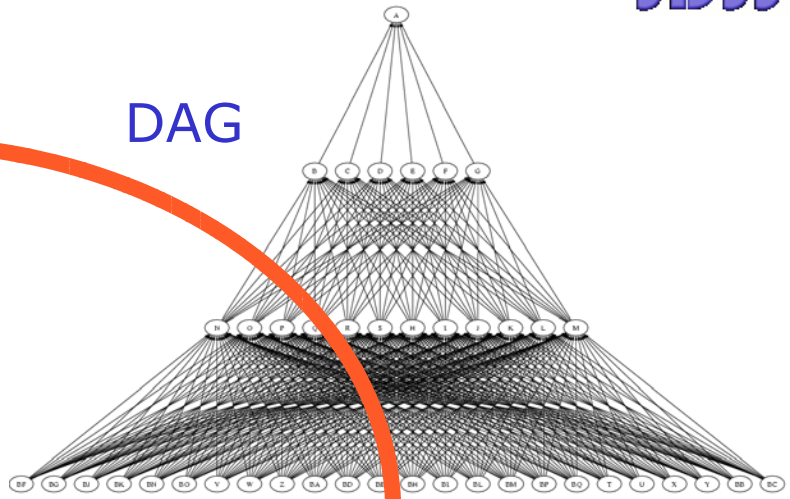
Work and slide by Rick Cavanaugh and Dimitri Bourilkov, University of Florida

Virtual Data Example: *Galaxy Cluster Search*

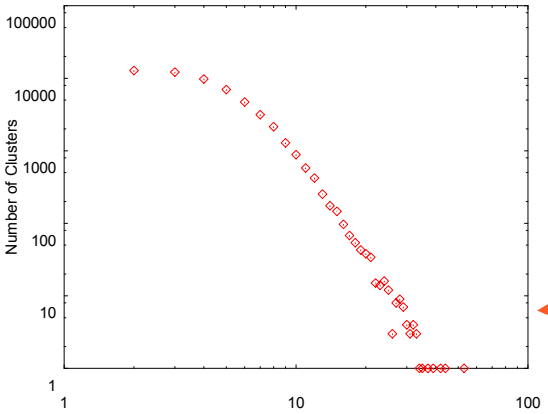


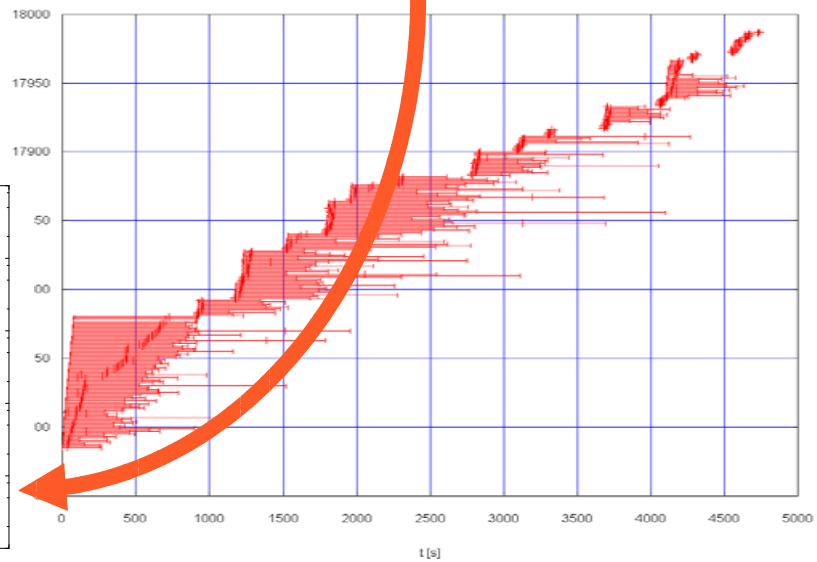
Sloan Data

DAG



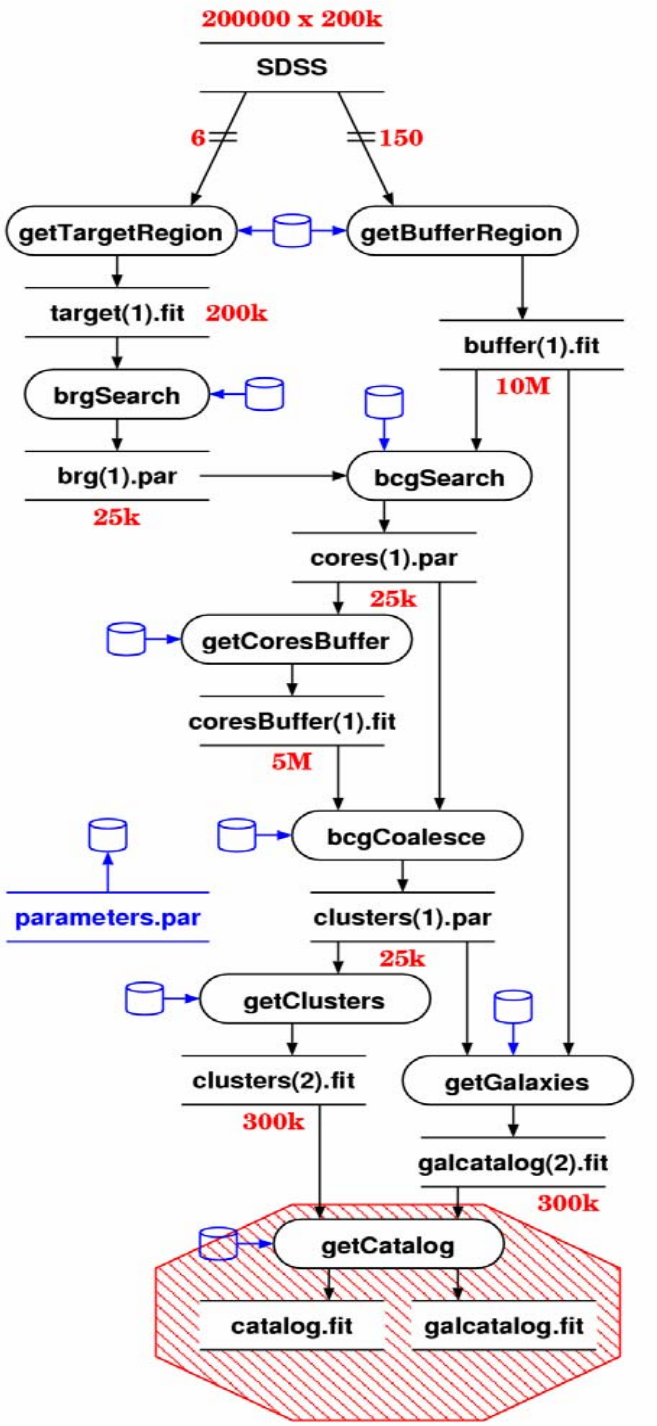
Galaxy cluster size distribution

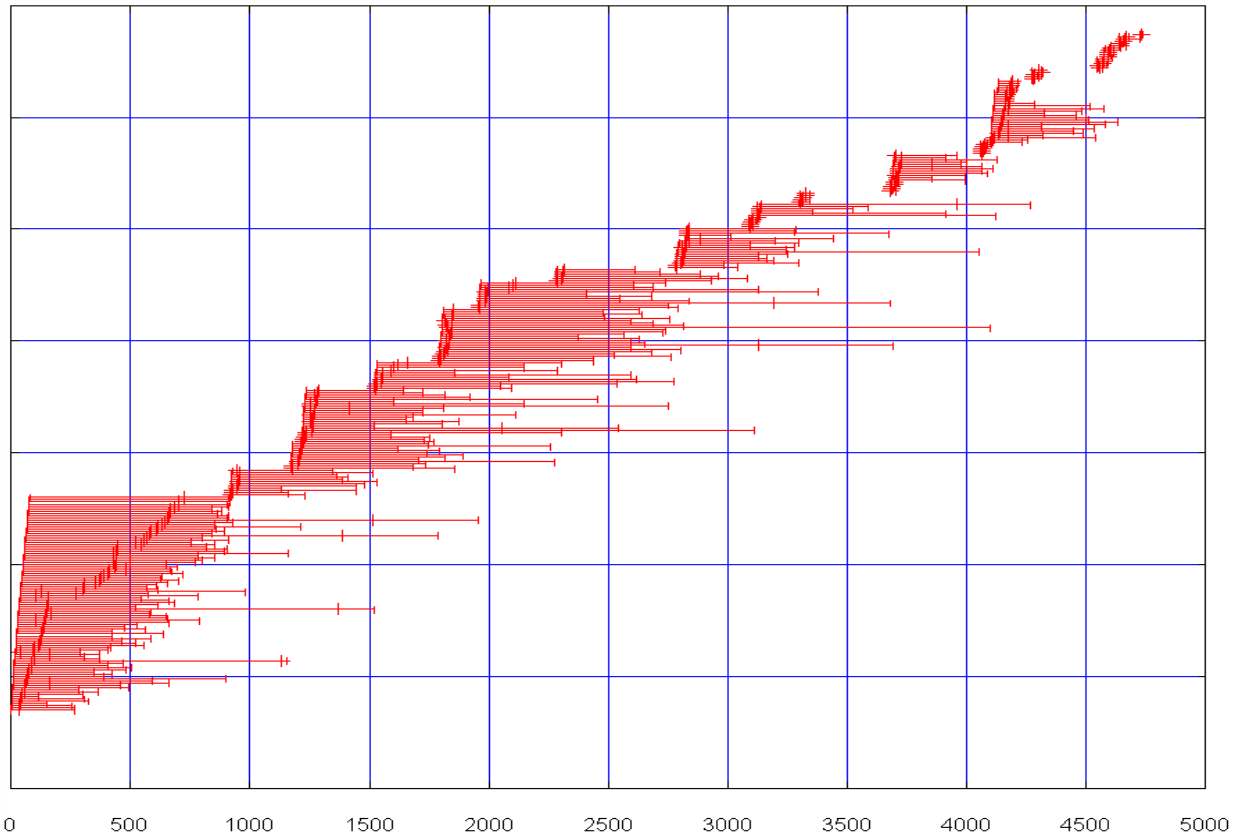




Jim Annis, Steve Kent, Vijay Sehkri, *Fermilab*, Michael Milligan, Yong Zhao, *University of Chicago*

Cluster Search Workflow Graph and Execution Trace

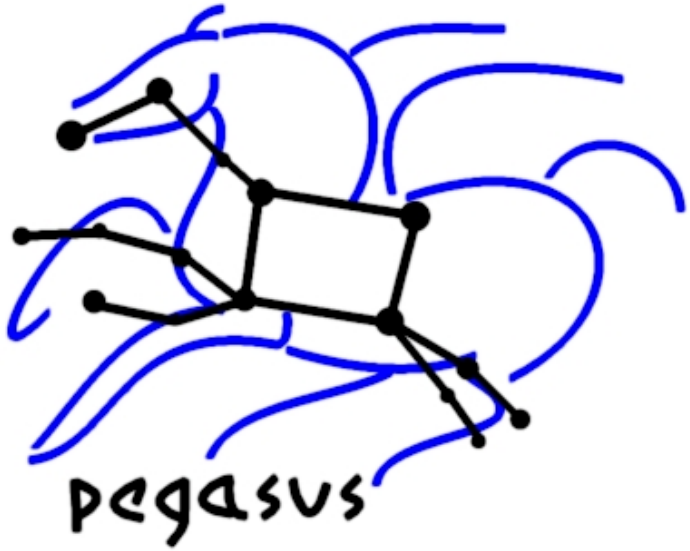




Workflow jobs vs time

Outline

- Workflows on the Grid
- The GriPhyN project
- Chimera
- Pegasus
- Research issues
- Exercises



the globus alliance
www.globus.org

Outline

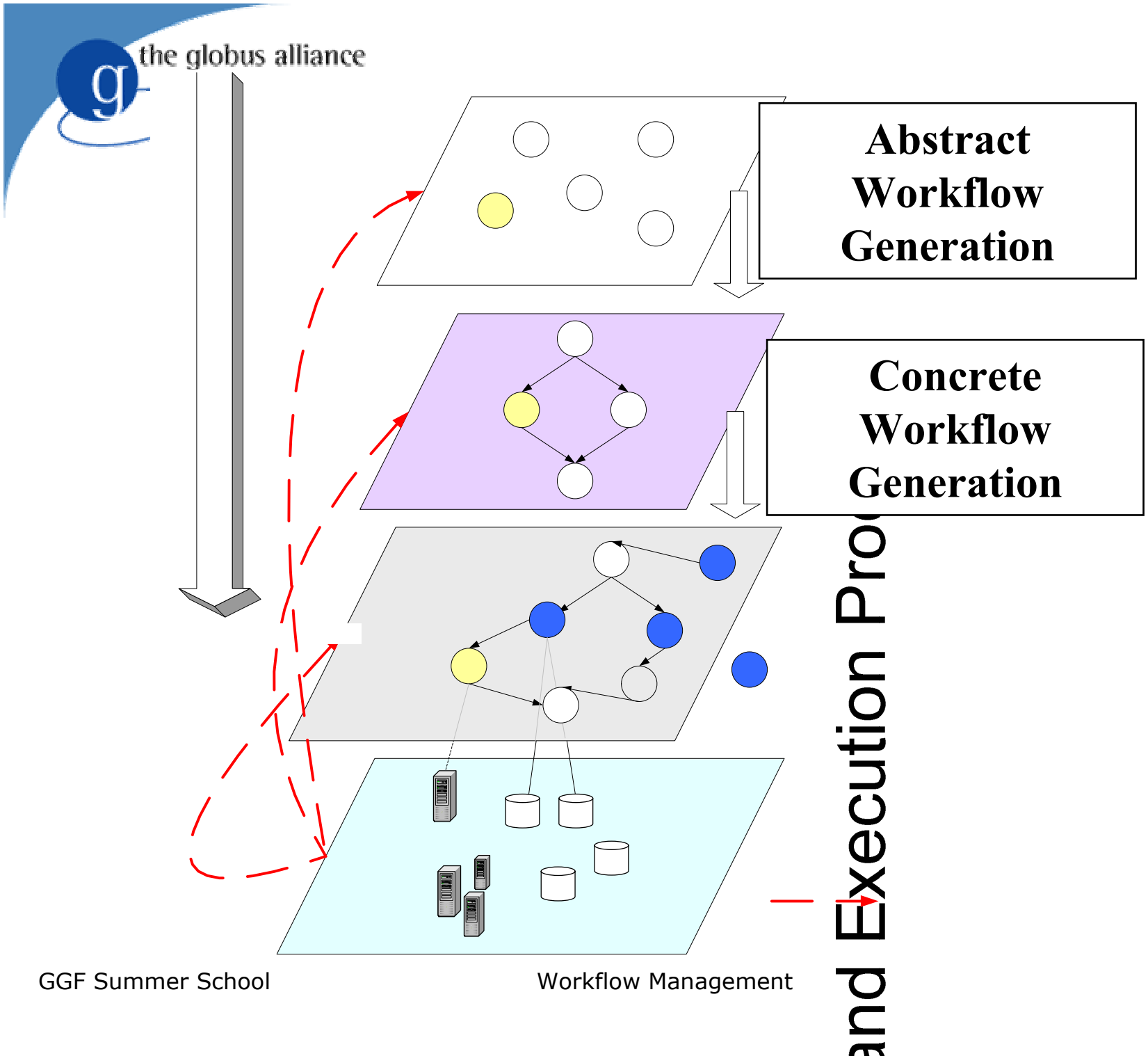
- Pegasus Introduction
- Pegasus and Other Globus Components
- Pegasus' Concrete Planner
- Deferred planning mode
- Pegasus portal
- Future Improvements

Grid Applications

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

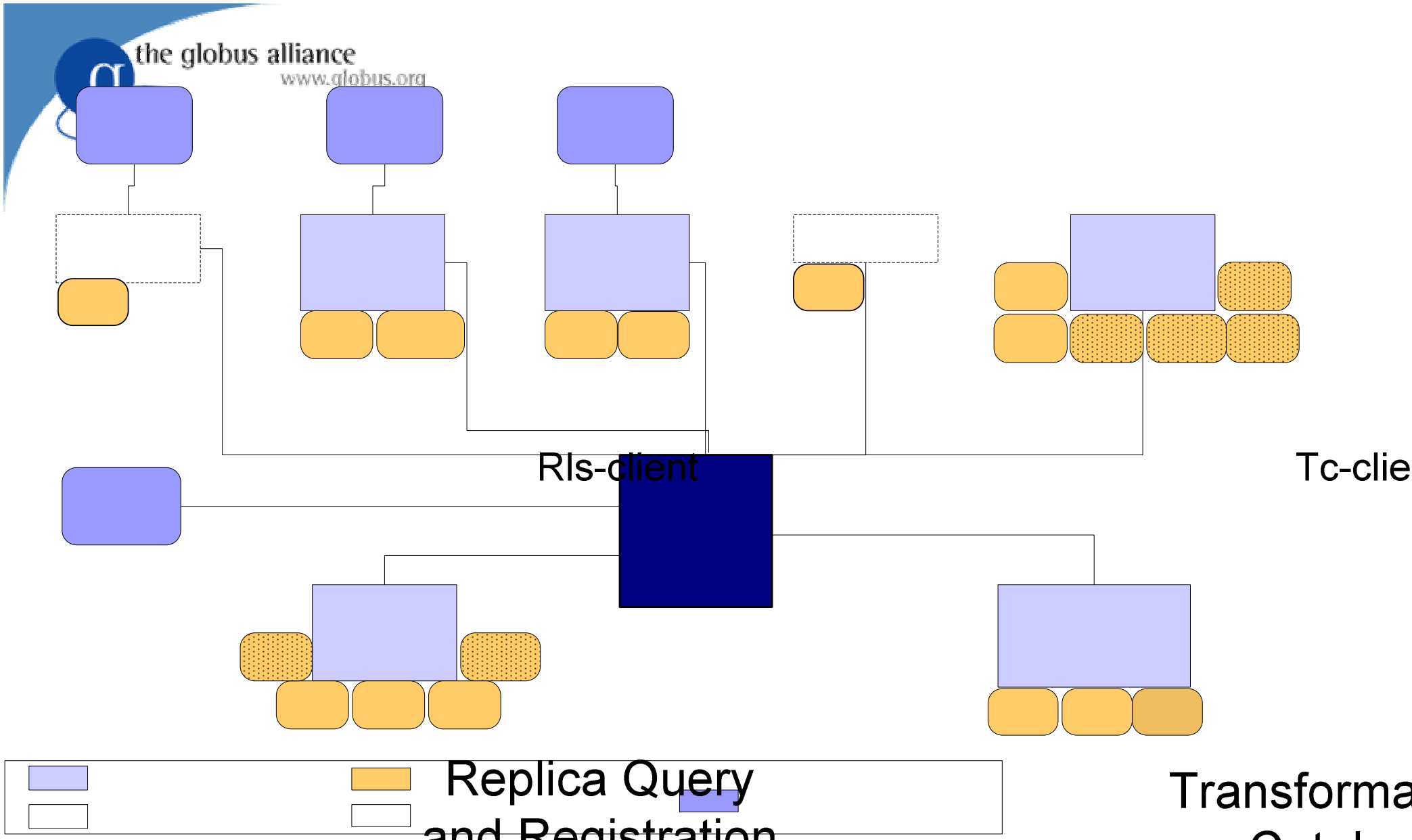
- Execution environment is complex and very dynamic
  - Resources come and go
  - Data is replicated
  - Components can be found at various locations or staged in on demand

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


the globus alliance
Abstract Workflow Generation
Concrete Workflow Generation
and Execution Pro

Pegasus

- Flexible framework, maps abstract workflows onto the Grid
- Possess well-defined APIs and clients for:
  - Information gathering
    - > Resource information
    - > Replica query mechanism
    - > Transformation catalog query mechanism
  - Resource selection
    - > Compute site selection
    - > Replica selection
  - Data transfer mechanism
- Can support a variety of workflow executors

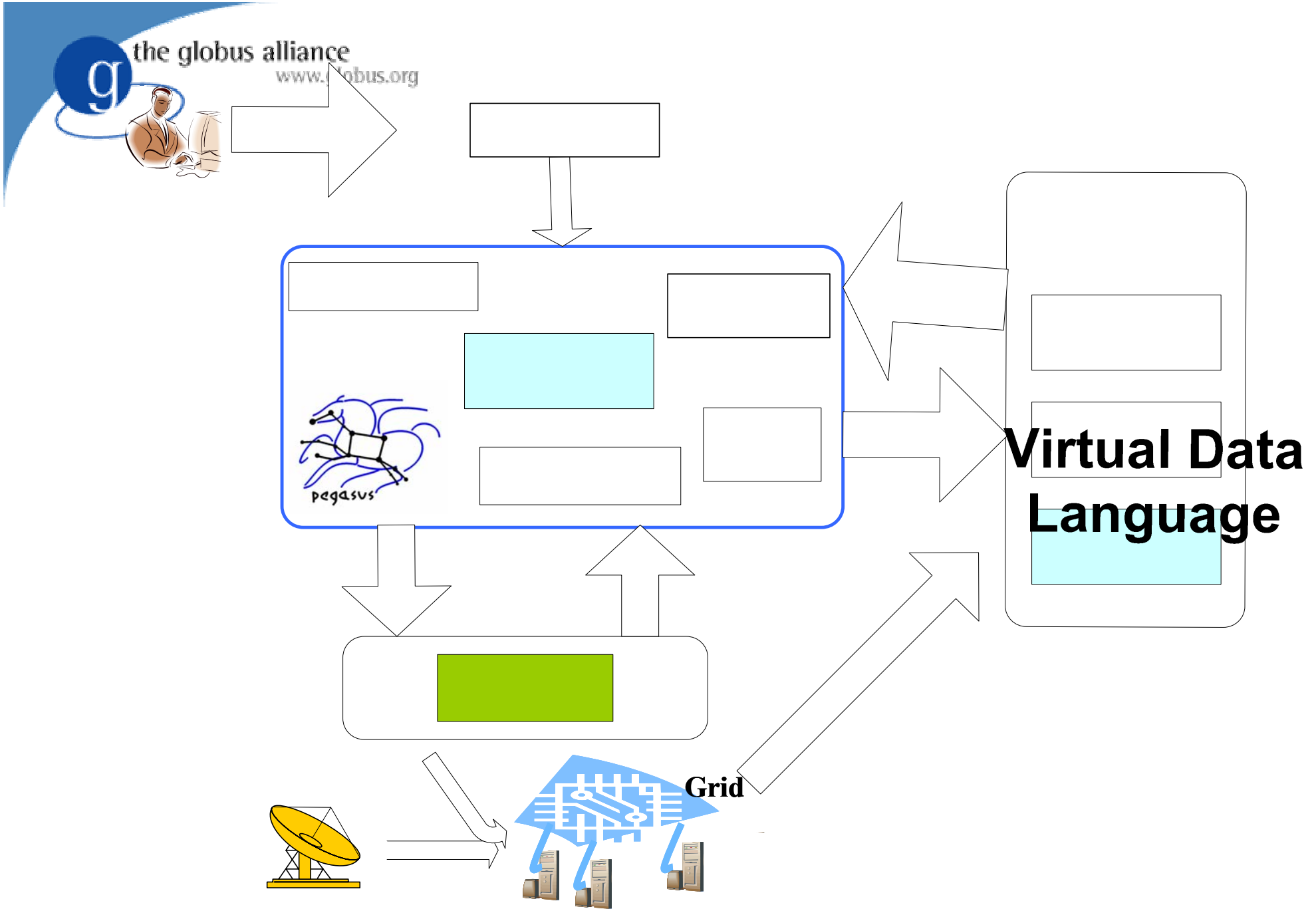


Pegasus Components

the globus alliance
www.globus.org

Pegasus: A particular configuration

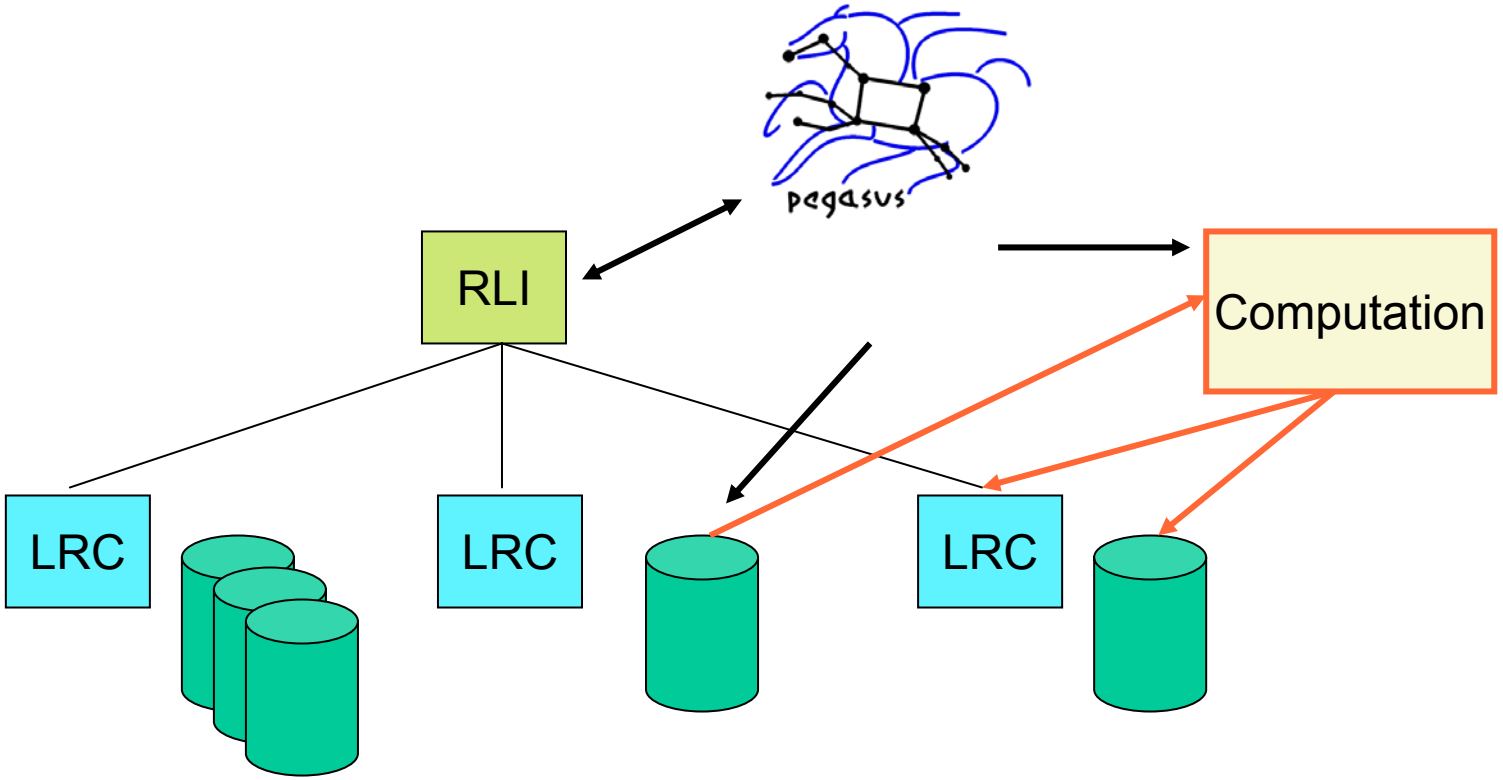
- Automatically locates physical locations for both components (transformations) and data
  - Use Globus RLS and the Transformation Catalog
- Finds appropriate resources to execute the jobs
  - Via Globus MDS
- Reuses existing data products where applicable
  - Possibly reduces the workflow
- Publishes newly derived data products
  - RLS, Chimera virtual data catalog


the globus alliance
www.globus.org
pegasus
Virtual Data
Language
Grid

Request M

Replica Location Service

- Pegasus uses the RLS to find input data

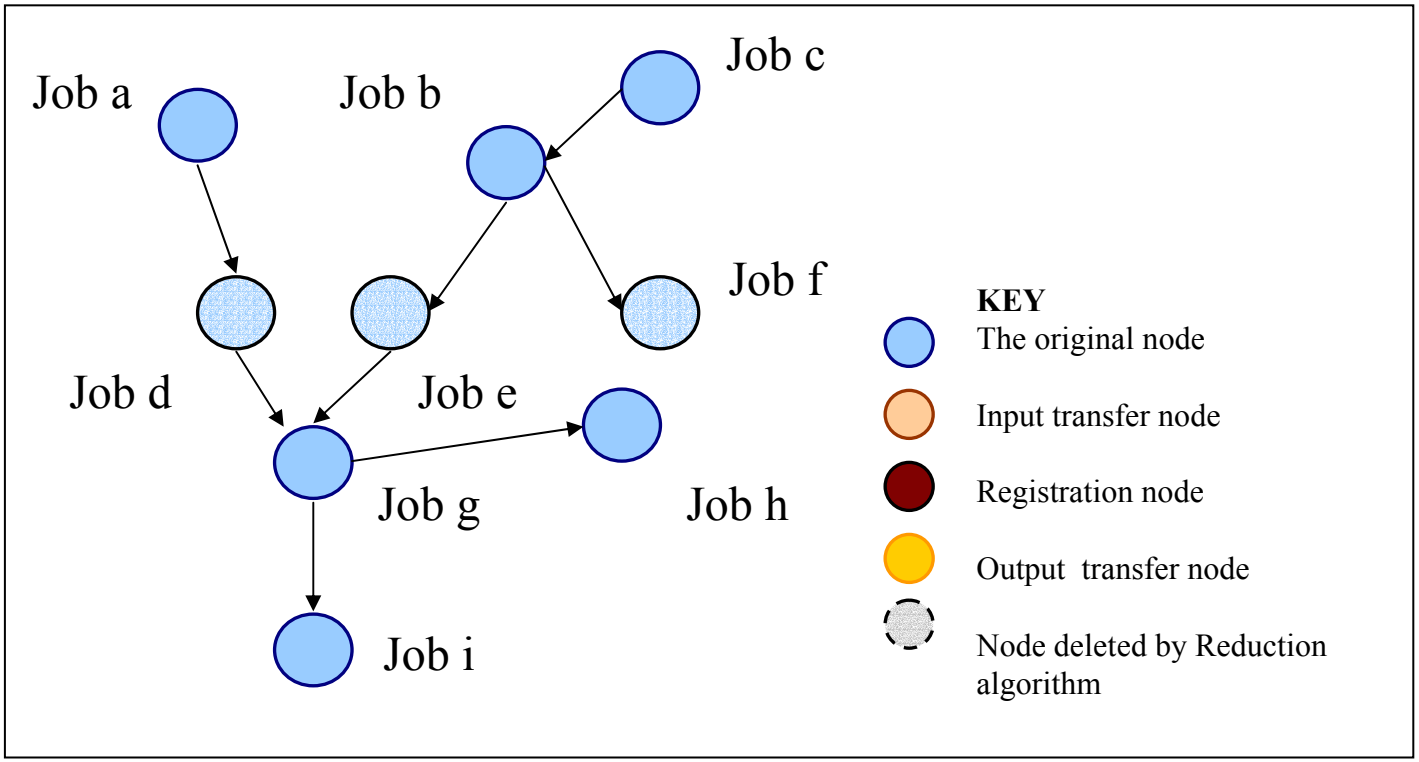


- Pegasus uses the RLS to register new data products

Use of MDS in Pegasus

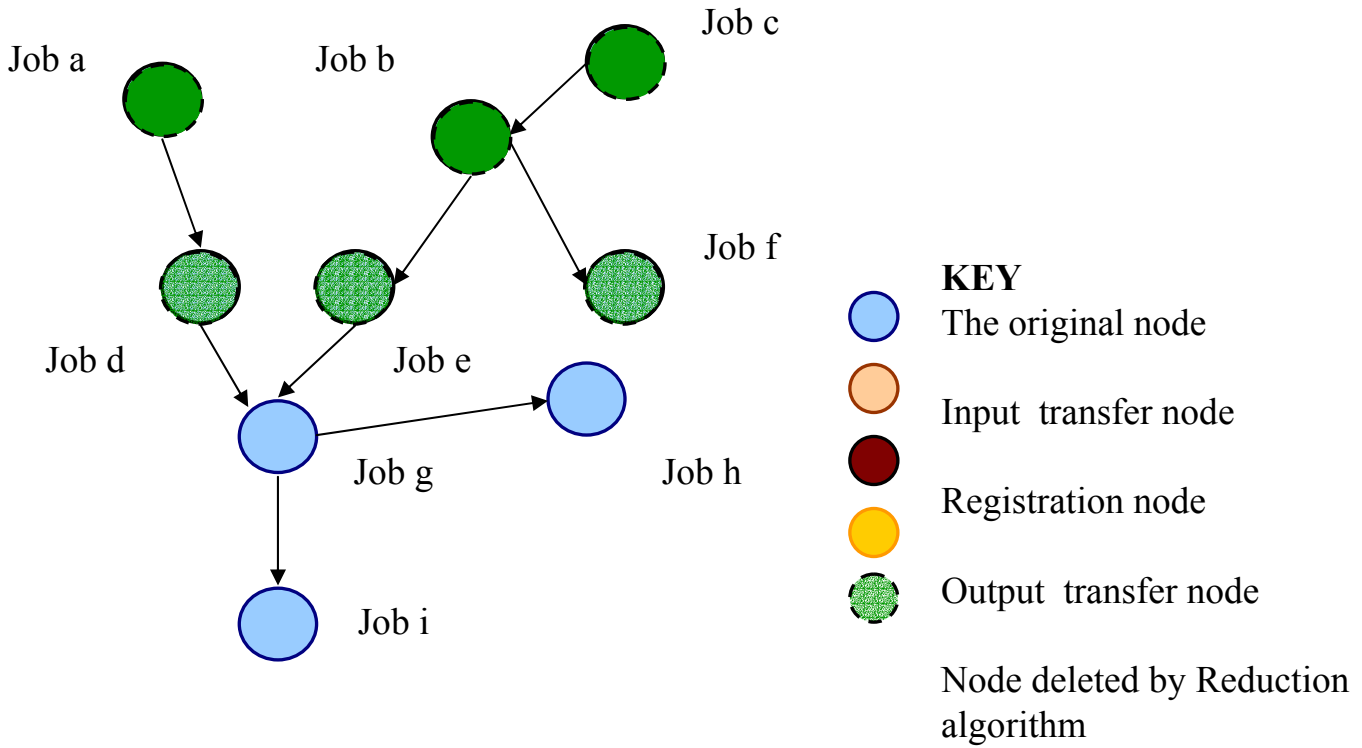
- MDS provides up-to-date Grid state information
  - Total and idle job queues length on a pool of resources (condor)
  - Total and available memory on the pool
  - Disk space on the pools
  - Number of jobs running on a job manager
- Can be used for resource discovery and selection
  - Developing various task to resource mapping heuristics
- Can be used to publish information necessary for replica selection
  - Developing replica selection components

Abstract Workflow Reduction



- The output jobs for the Dag are all the leaf nodes i.e. **f, h, i**.
- Each job requires 2 input files and generates 2 output files.
- The user specifies the output location

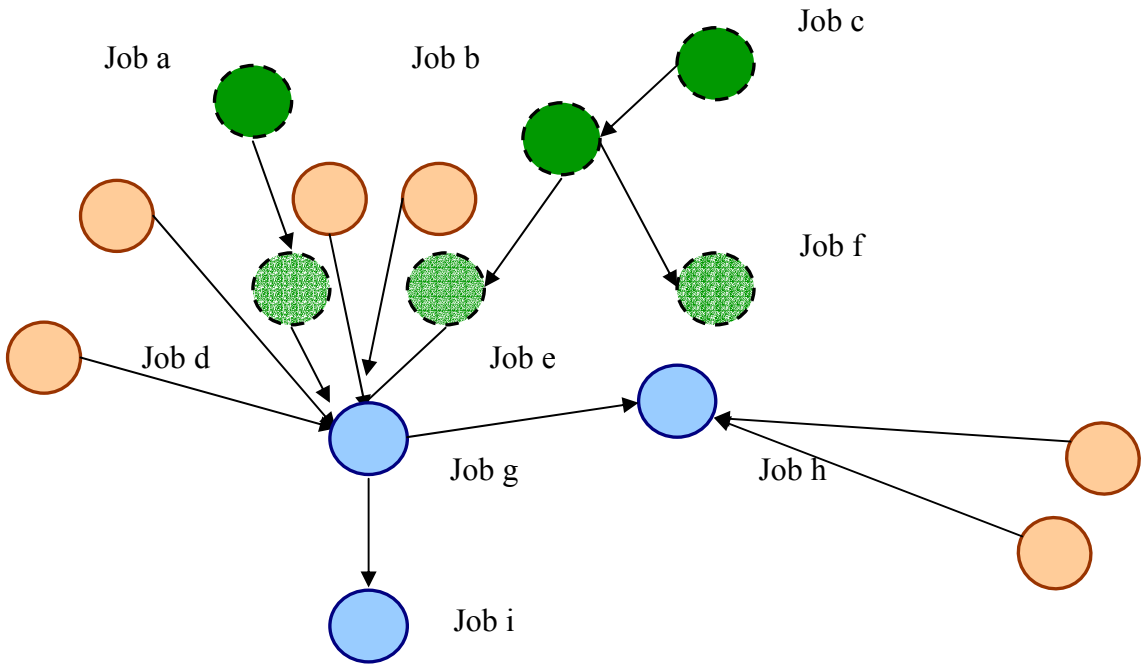
Optimizing from the point of view of Virtual Data



- Jobs **d, e, f** have output files that have been found in the Replica Location Service.
- Additional jobs are deleted.
- All jobs (**a, b, c, d, e, f**) are removed from the DAG.

Planner picks execution and replica locations

Plans for staging data in

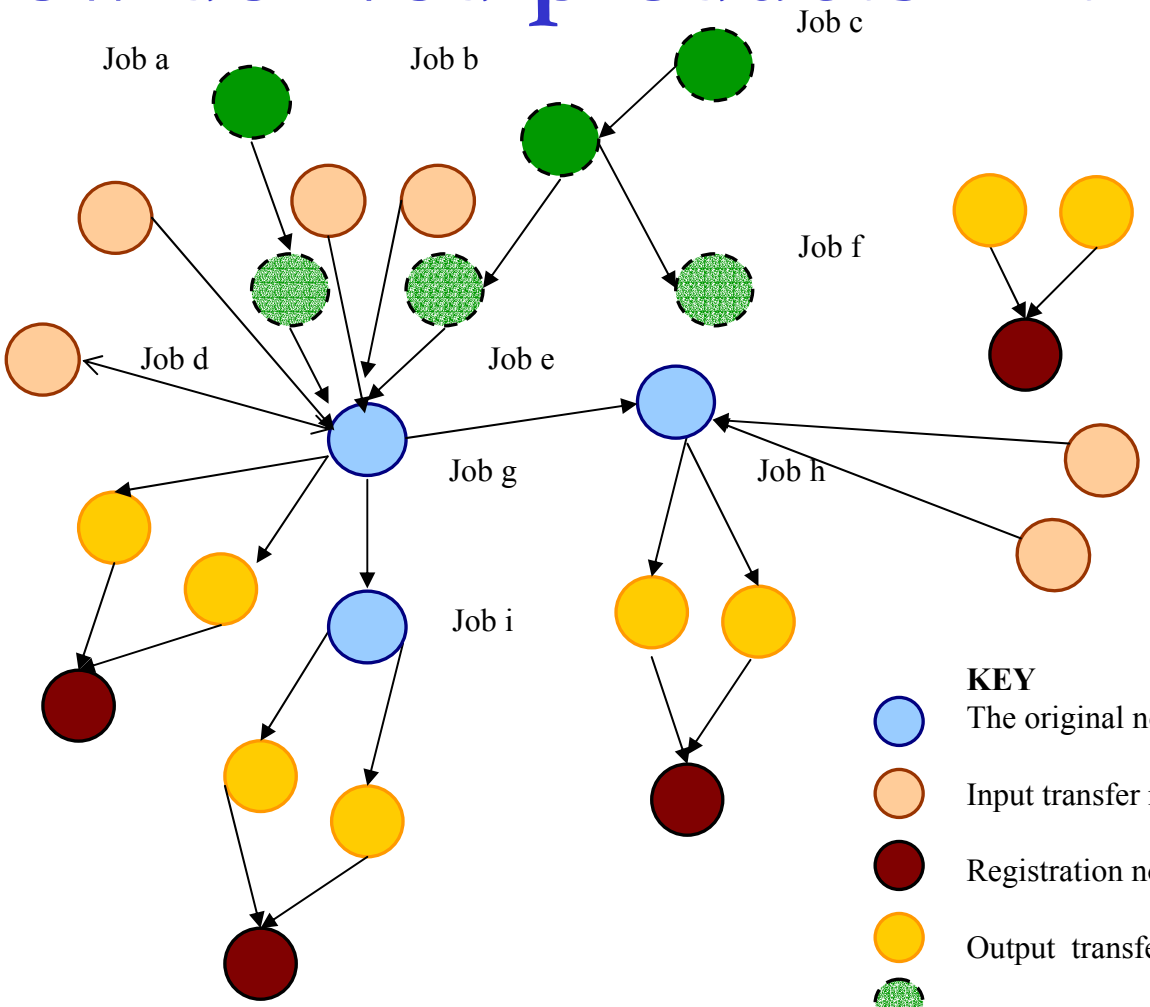


adding transfer nodes for the input files for the root nodes

KEY

- The original node
- Input transfer node
- Registration node
- Output transfer node
- Node deleted by Reduction algorithm

Staging data out and registering new derived products in the RLS

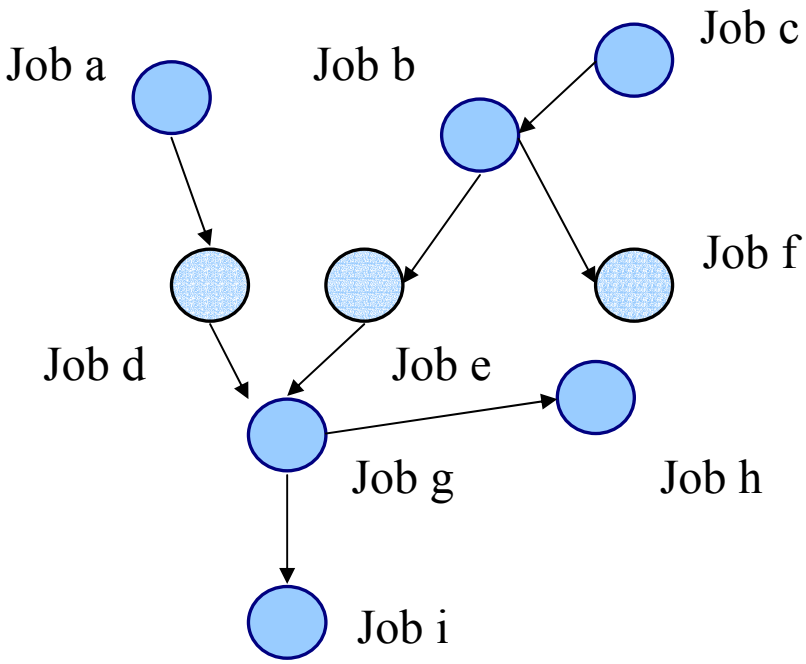


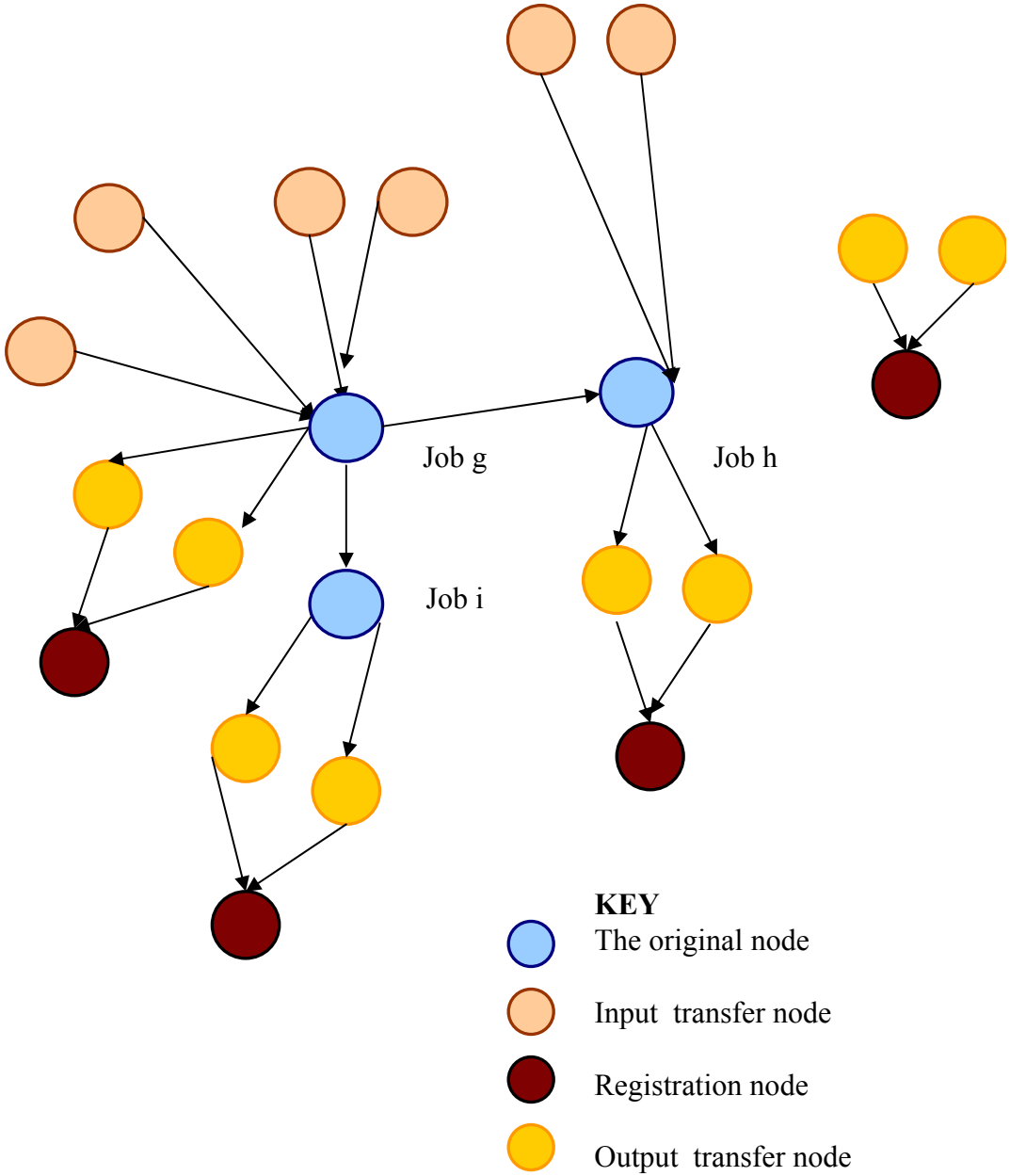
transferring the output files of the leaf job (**f**) to the output location

Staging and registering for each job that materializes data (**g, h, i**).

The final, executable DAG

Input DAG





Pegasus Components

- Concrete Planner and Submit file generator (gencdag)
  - The Concrete Planner of the VDS makes the logical to physical mapping of the DAX taking into account the pool where the jobs are to be executed (execution pool) and the final output location (output pool).
- Java Replica Location Service Client (rls-client & rls-query-client)
  - Used to populate and query the globus replica location service.

Pegasus Components (cont'd)

- XML Pool Config generator (genpoolconfig)
  - The Pool Config generator queries the MDS as well as local pool config files to generate a XML pool config which is used by Pegasus.
  - MDS is preferred for generation pool configuration as it provides a much richer information about the pool including the queue statistics, available memory etc.
- The following catalogs are looked up to make the translation
  - Transformation Catalog (tc.data)
  - Pool Config File
  - Replica Location Services
  - Monitoring and Discovery Services

Transformation Catalog (Demo)

- Consists of a simple text file.
  - Contains Mappings of Logical Transformations to Physical Transformations.
- Format of the tc.data file

```
#poolname   logical tr      physical tr                    env
isi             preprocess    /usr/vds/bin/preprocess
   VDS_HOME=/usr/vds/;
```

- All the physical transformations are absolute path names.
- Environment string contains all the environment variables required in order for the transformation to run on the execution pool.
- DB based TC in testing phase.

Pool Config (Demo)

- Pool Config is an XML file which contains information about various pools on which DAGs may execute.
- Some of the information contained in the Pool Config file is
  - Specifies the various job-managers that are available on the pool for the different types of condor universes.
  - Specifies the GridFtp storage servers associated with each pool.
  - Specifies the Local Replica Catalogs where data residing in the pool has to be cataloged.
  - Contains profiles like environment hints which are common site-wide.
  - Contains the working and storage directories to be used on the pool.

the globus alliance
www.globus.org

Pool config

- Two Ways to construct the Pool Config File.
  - Monitoring and Discovery Service
  - Local Pool Config File (Text Based)
- Client tool to generate Pool Config File
  - The tool genpoolconfig is used to query the MDS and/or the local pool config file/s to generate the XML Pool Config file.

Gvds.Pool.Config

- This file is read by the information provider and published into MDS.
- Format

  gvds.pool.id : <POOL ID>

  gvds.pool.lrc : <LRC URL>

  gvds.pool.gridftp : <GSIFTP URL>@<GLOBUS VERSION>

  gvds.pool.gridftp : gsiftp://sukhna.isi.edu/nfs/asd2/gmehta@2.4.0

  gvds.pool.universe : <UNIVERSE>@<JOBMANAGER URL>@<GLOBUS VERSION>

  gvds.pool.universe : transfer@columbus.isi.edu/jobmanager-fork@2.2.4

  gvds.pool.gridlaunch : <Path to Kickstart executable>

  gvds.pool.workdir : <Path to Working Dir>

  gvds.pool.profile : <namespace>@<key>@<value>

  gvds.pool.profile : env@GLOBUS_LOCATION@/smarty/gt2.2.4

  gvds.pool.profile : vds@VDS_HOME@/nfs/asd2/gmehta/vds

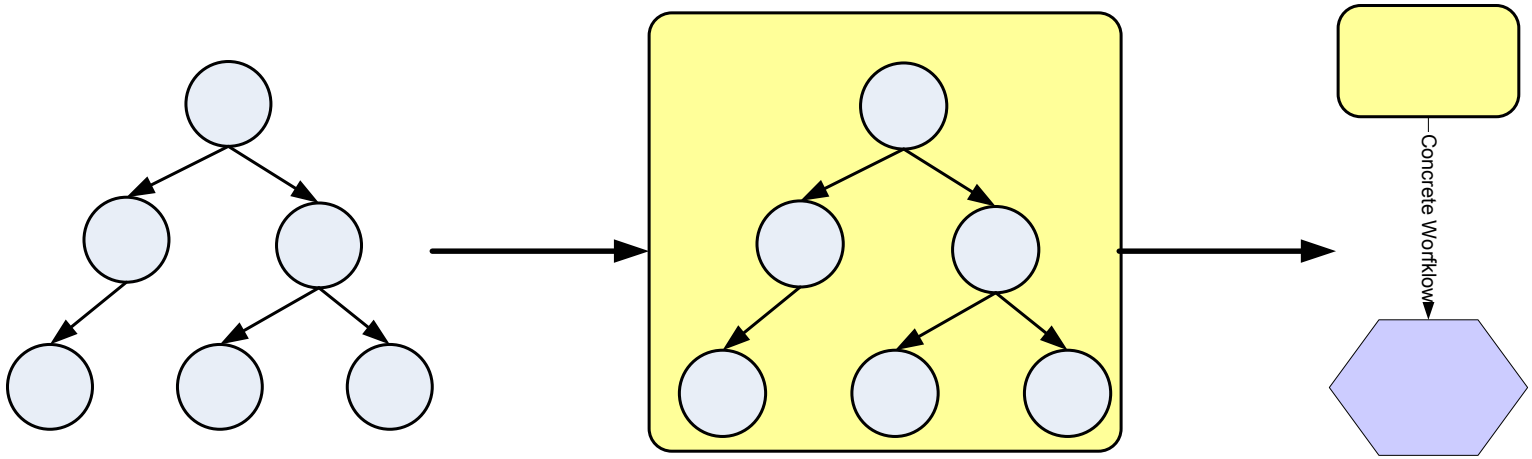
Properties

- Properties file define and modify the behavior of Pegasus.
- Properties set in the $VDS_HOME/properties can be overridden by defining them either in $HOME/.chimerarc or by giving them on the command line of any executable.
  - eg. Gendax –Dvds.home=path to vds home……
- Some examples follow but for more details please read the sample.properties file in $VDS_HOME/etc directory.
- Basic Required Properties
  - vds.home : This is auto set by the clients from the environment variable $VDS_HOME
  - vds.properties : Path to the default properties file
    - Default : ${vds.home}/etc/properties

Concrete Planner Gencdag

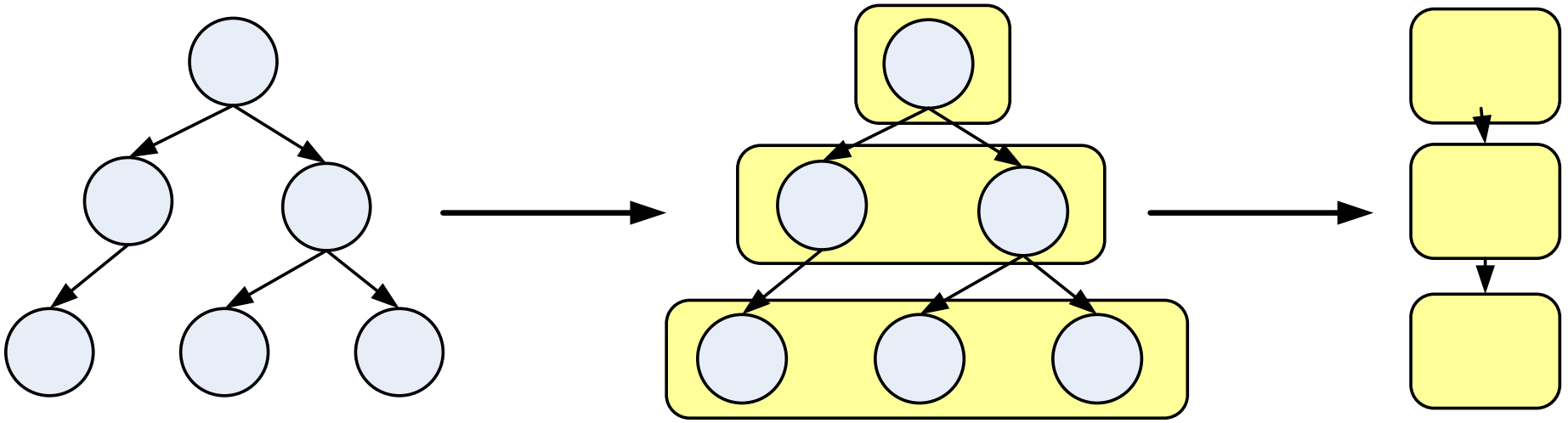
- The Concrete planner takes the DAX produced by Chimera and converts into a set of condor dag and submit files.
- Usage : gencdag --dax <dax file> --p <list of execution pools> [--dir <dir for o/p files>] [--o <outputpool>] [--force]
- You can specify more then one execution pools. Execution will take place on the pools on which the executable exists. If the executable exists on more then one pool then the pool on which the executable will run is selected randomly.
- Output pool is the pool where you want all the output products to be transferred to. If not specified the materialized data stays on the execution pool

Original Pegasus configuration



Simple scheduling: random or round robin using well-defined scheduling interfaces.

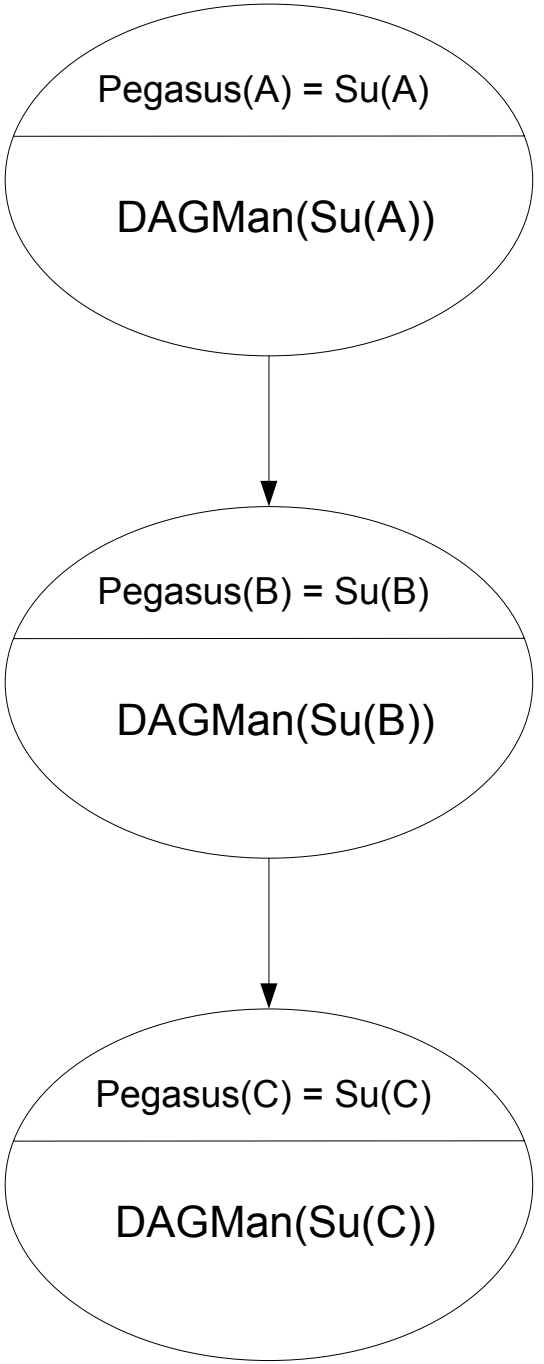
Deferred Planning through Partitioning



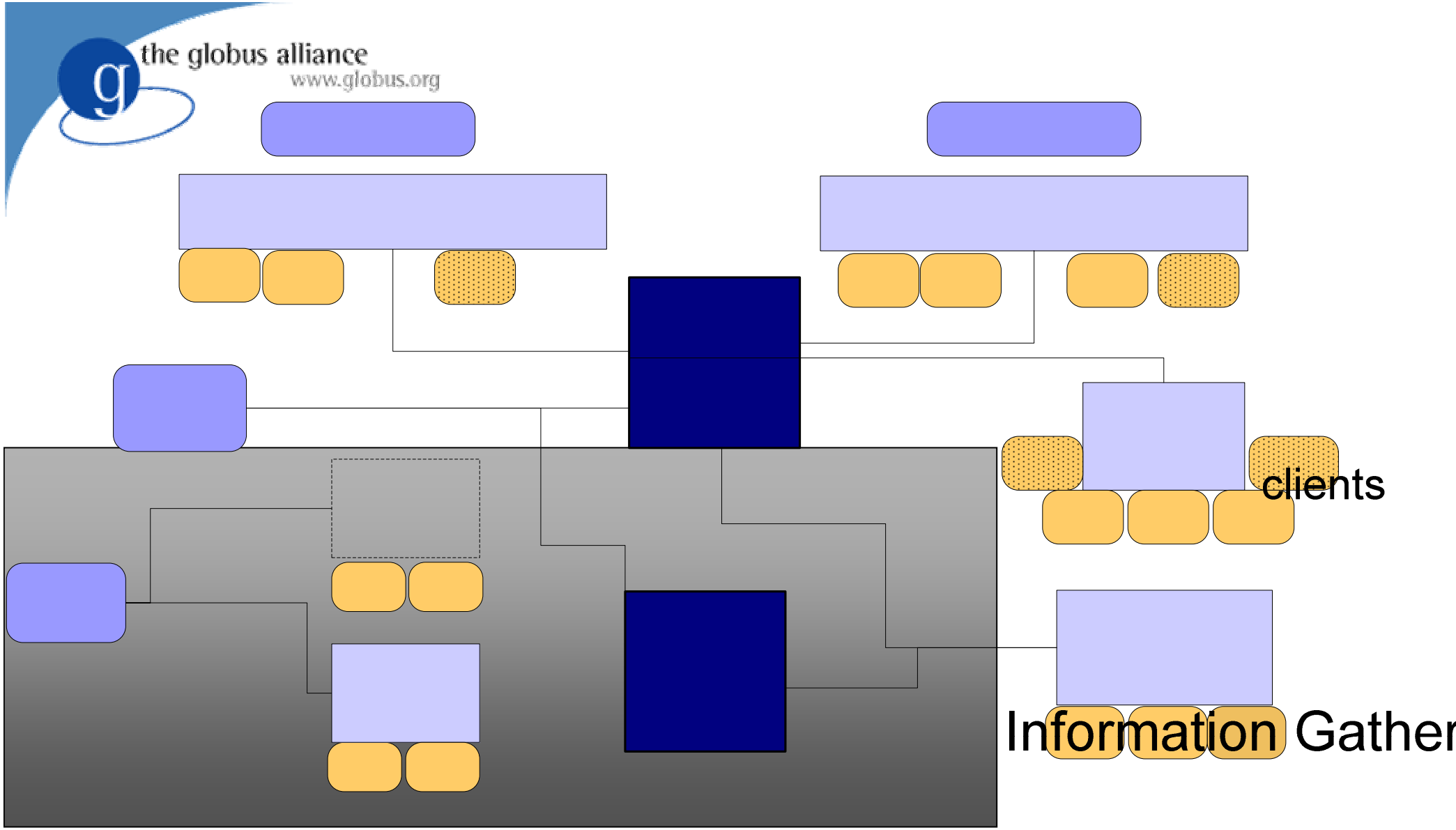
A variety of planning algorithms can be implemented

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

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

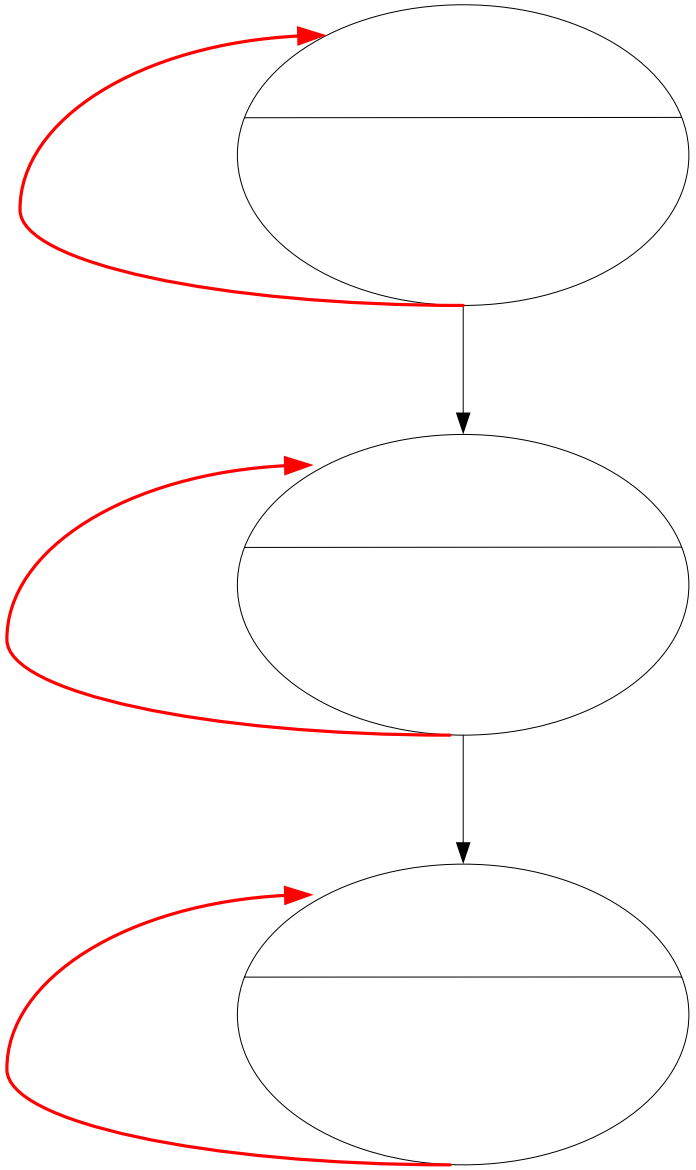


Mega DAG is created by Pegasus and then submitted to DAGMan



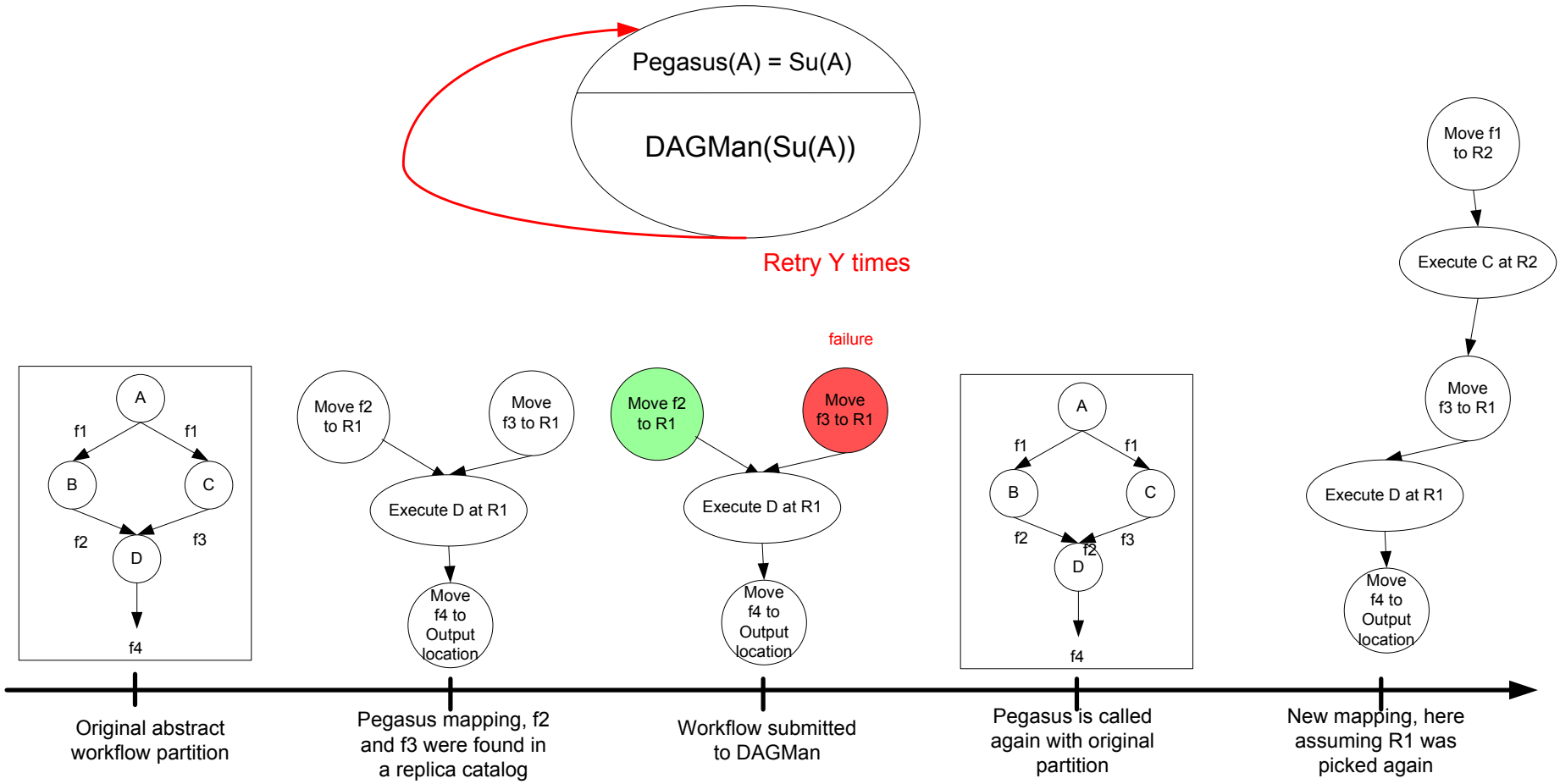
Mega DAG Pegasus

Re-planning capabilities



Pegasus

Complex Replanning for Free (almost)



Optimizations

- If the workflow being refined by Pegasus consists of only 1 node
  - Create a condor submit node rather than a dagman node
  - This optimization can leverage Euryale's super-node writing component

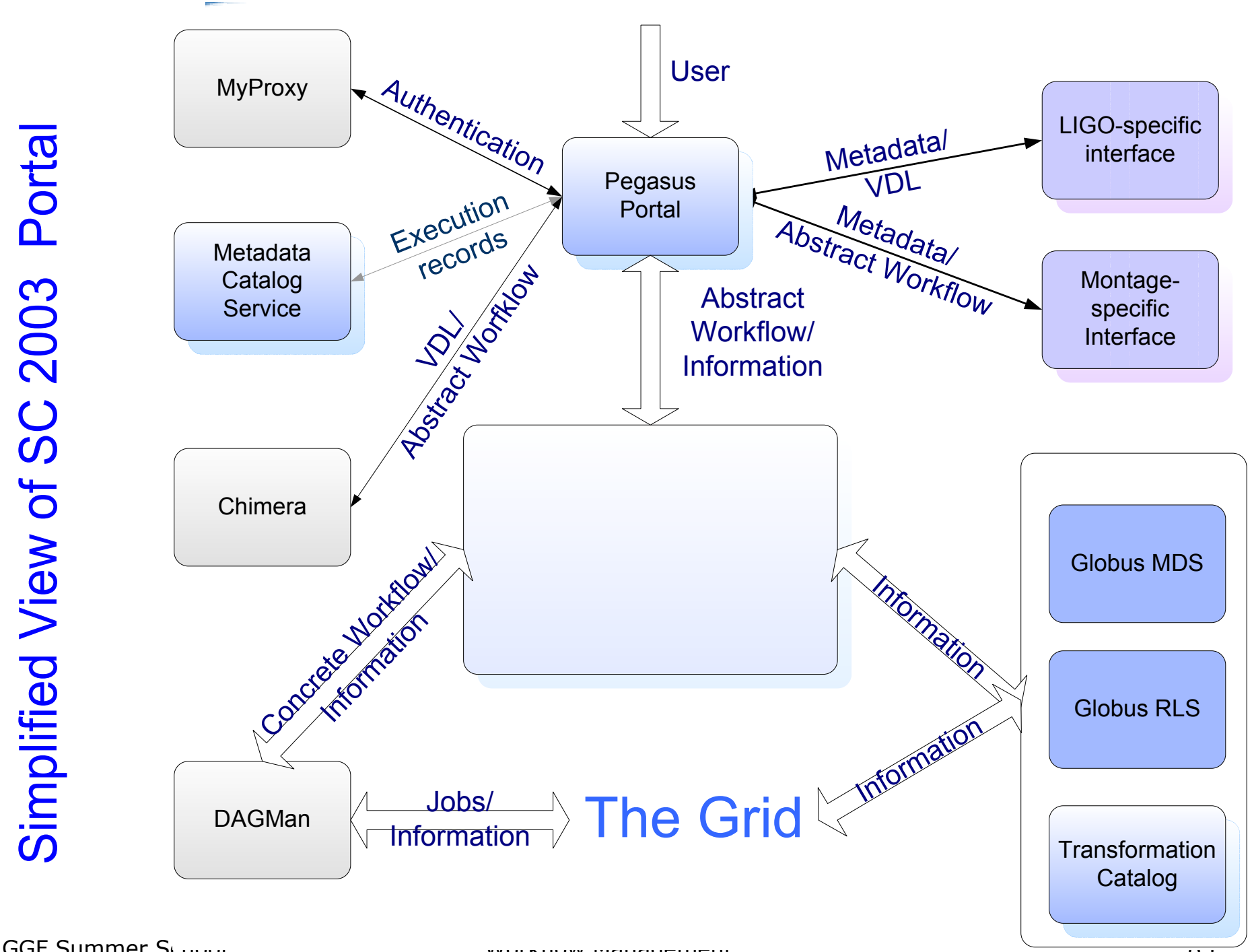
Planning & Scheduling Granularity

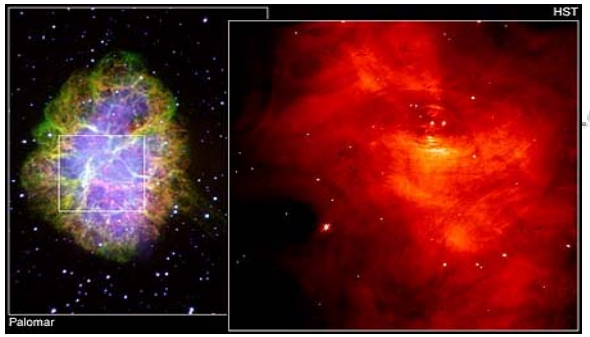
- Partitioning
  - Allows to set the granularity of planning ahead
- Node aggregation
  - Allows to combine nodes in the workflow and schedule them as one unit (minimizes the scheduling overheads)
  - May reduce the overheads of making scheduling and planning decisions
- Related but separate concepts
  - Small jobs
    - > High-level of node aggregation
    - > Large partitions
  - Very dynamic system
    - > Small partitions

- Create workflow partitions
  - partitiondax --dax ./blackdiamond.dax --dir dax
- Create the MegaDAG (creates the dagman submit files)
  - gencdag - Dvds.properties=~/conf/properties --pdax ./dax/blackdiamond.pdax --pools isi_condor --o isi_condor --dir ./dags/

  Note the --pdax option instead of the normal --dax option.
- submit the .dag file for the mega dag
  - condor_submit_dag black-diamond_0.dag

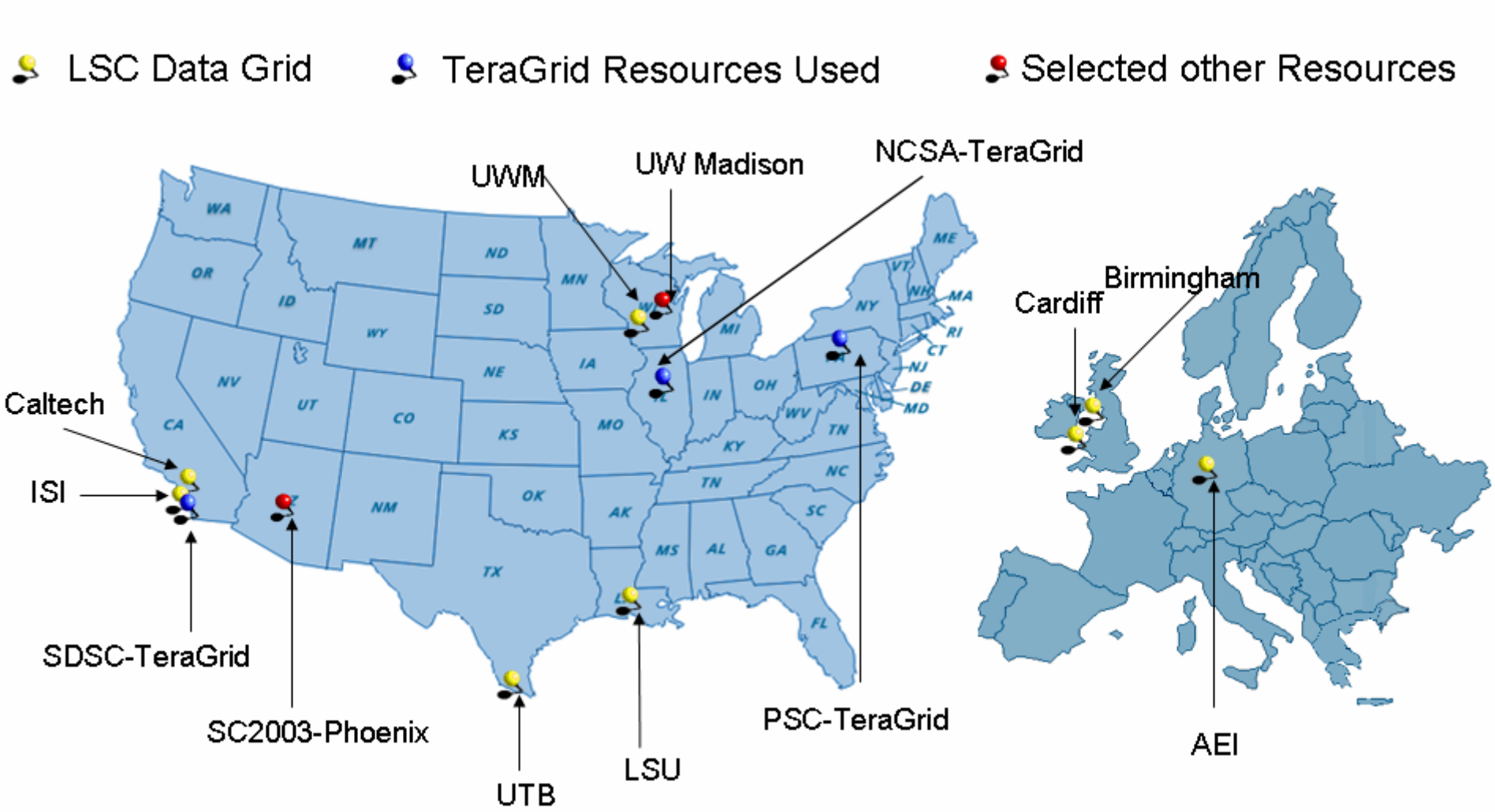
Simplified View of SC 2003 Portal





LIGO Scientific Collaboration

- Continuous gravitational waves are expected to be produced by a variety of celestial objects
- Only a small fraction of potential sources are known
- Need to perform blind searches, scanning the regions of the sky where we have no a priori information of the presence of a source
  - Wide area, wide frequency searches
- Search is performed for potential sources of continuous periodic waves near the Galactic Center and the galactic core
- The search is very compute and data intensive
- LSC used the occasion of SC2003 to initiate a month-long production run with science data collected during 8 weeks in the Spring of 2003



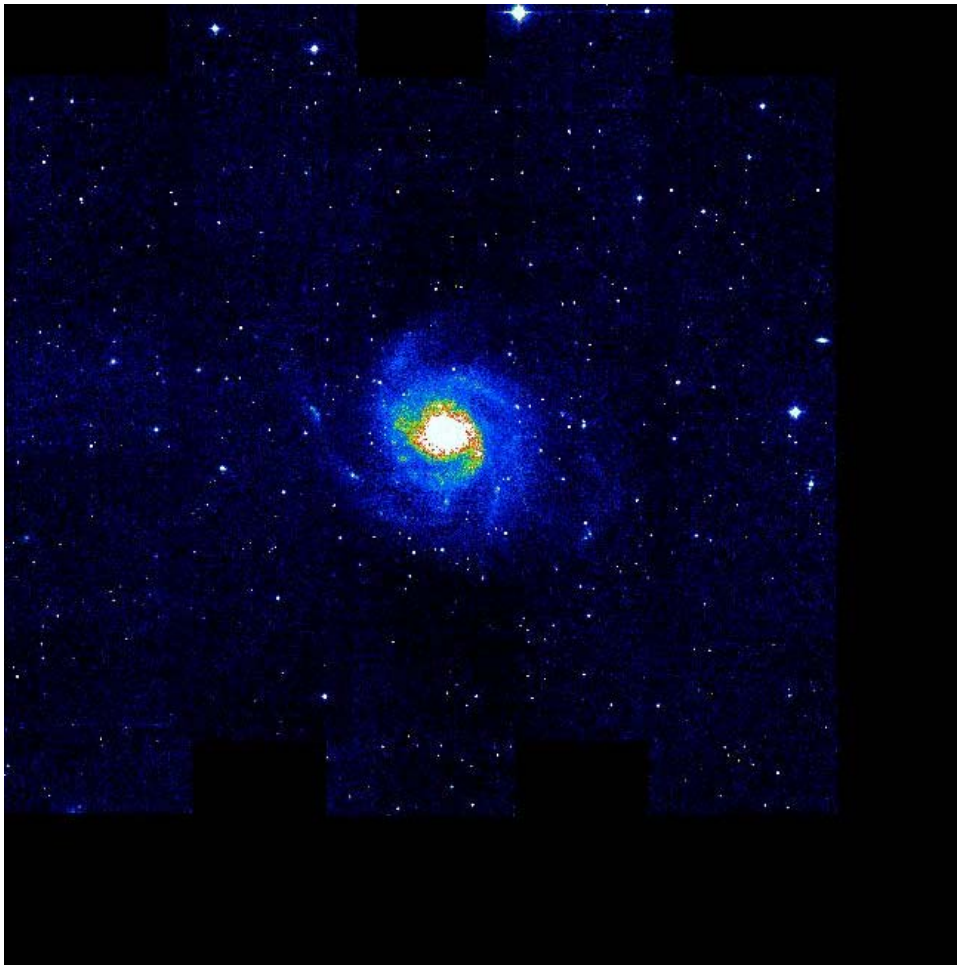
Additional resources used: Grid3 iVDGL resources

LIGO Acknowledgements

- Bruce Allen, Scott Koranda, Brian Moe, Xavier Siemens, University of Wisconsin Milwaukee, USA
- Stuart Anderson, Kent Blackburn, Albert Lazzarini, Dan Kozak, Hari Pulapaka, Peter Shawhan, Caltech, USA
- Steffen Grunewald, Yousuke Itoh, Maria Alessandra Papa, Albert Einstein Institute, Germany
- Many Others involved in the Testbed
- www.ligo.caltech.edu
- www.lsc- group.phys.uwm.edu/lscdatagrid/
- http://pandora.aei.mpg.de/merlin/
- LIGO Laboratory operates under NSF cooperative agreement PHY-0107417

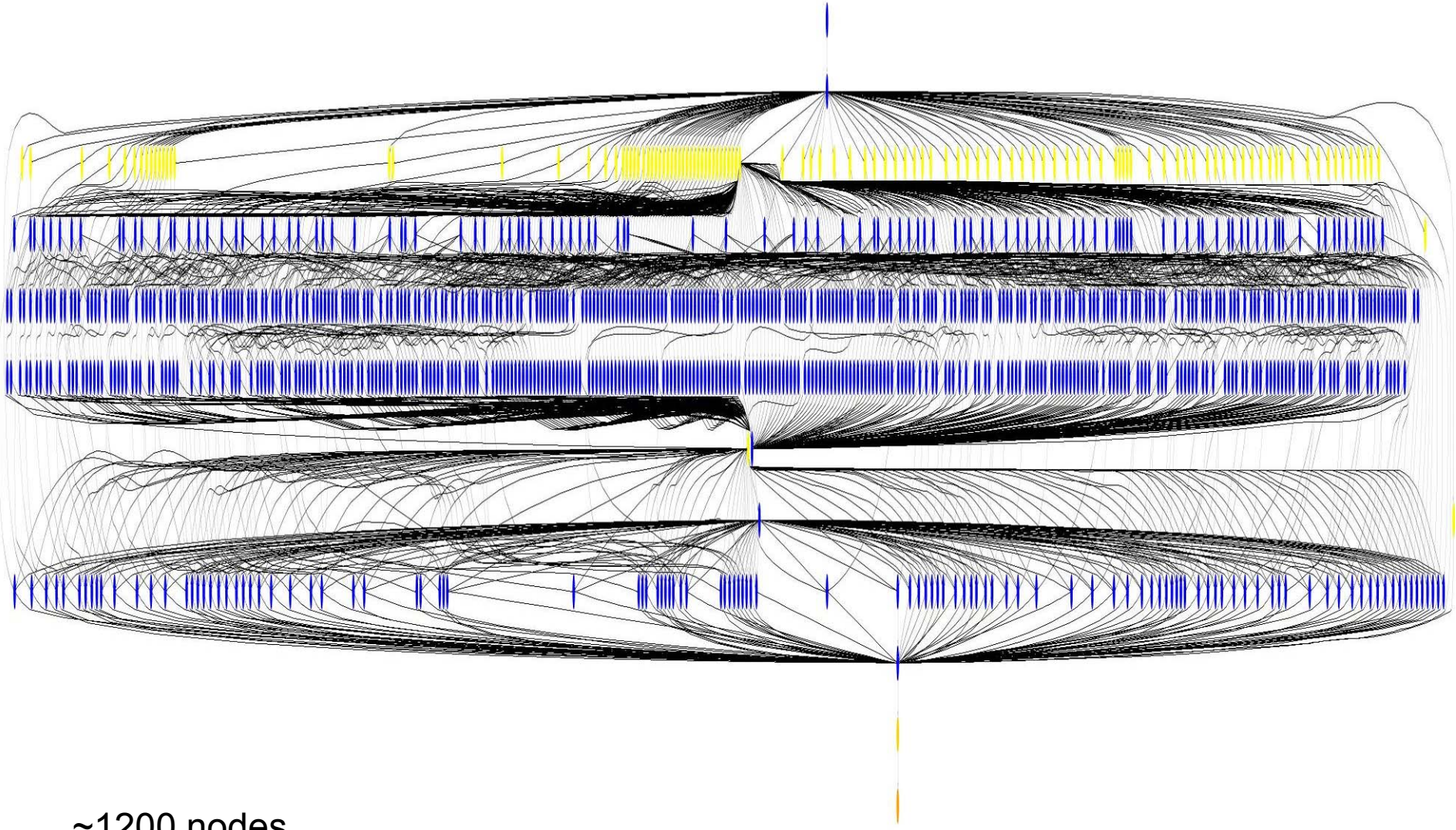
Montage

- 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.

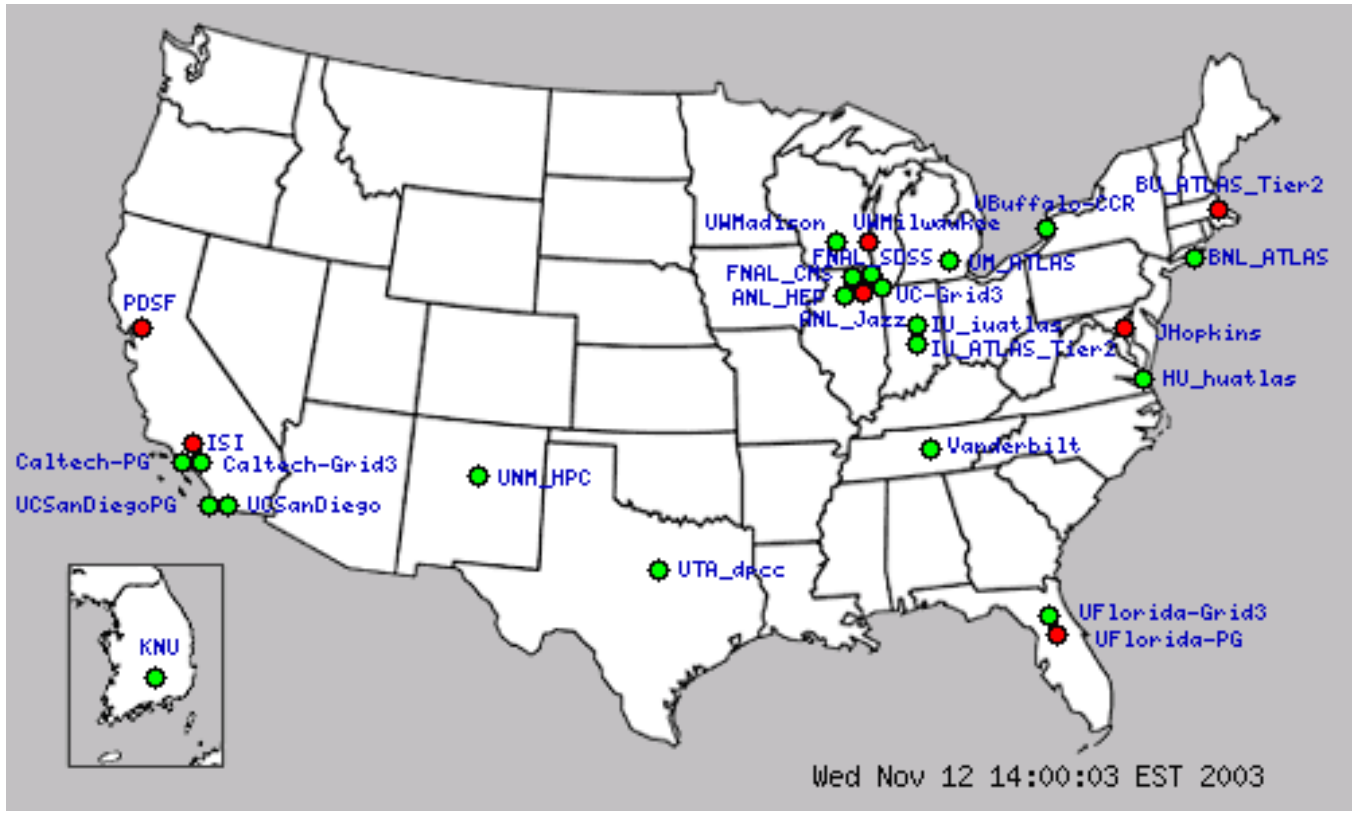
Portal Demonstration

the globus alliance
www.globus.org

Outline

- Workflows on the Grid
- The GriPhyN project
- Chimera
- Pegasus
- Research issues
- Exercises

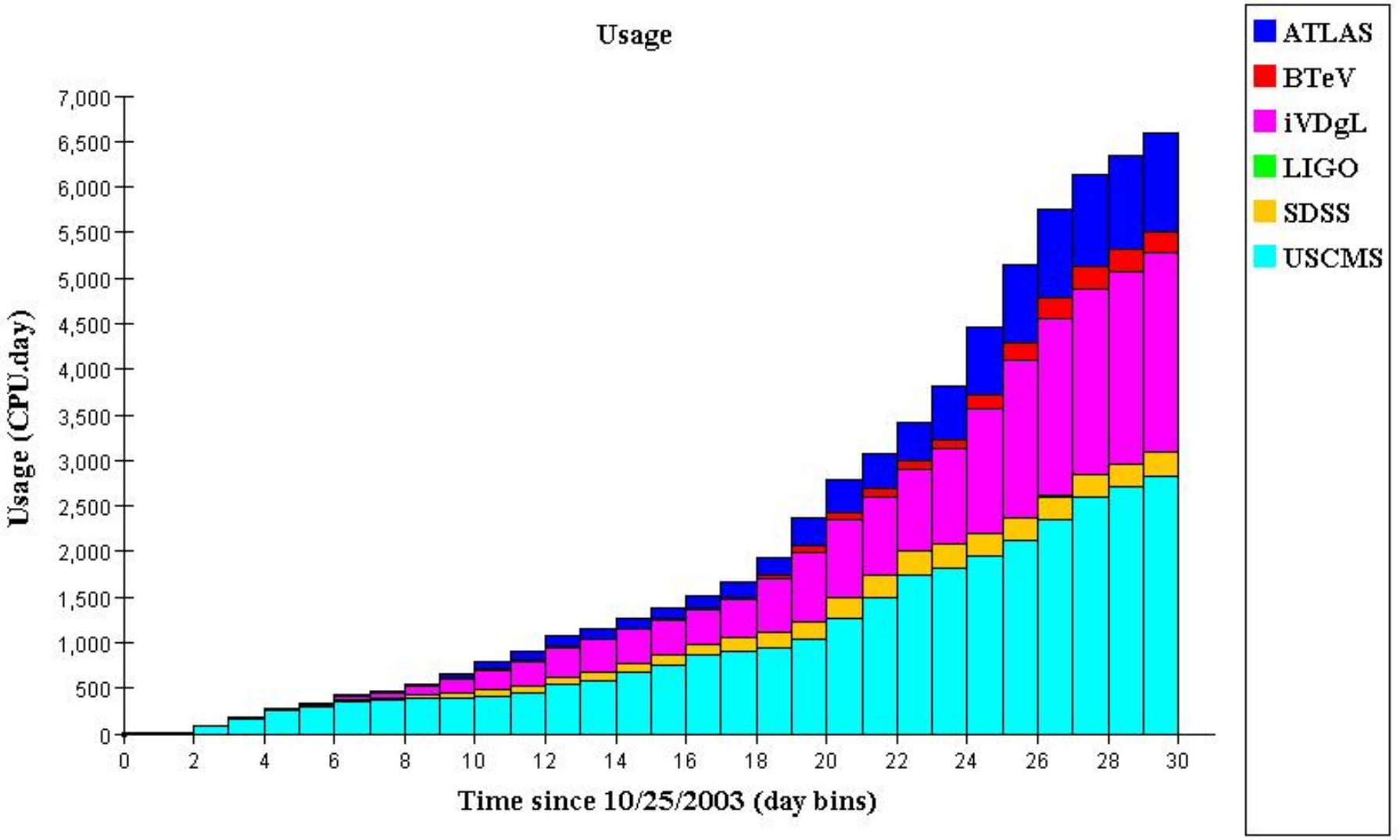
Grid3 – The Laboratory



Supported by the National Science Foundation and the Department of Energy.

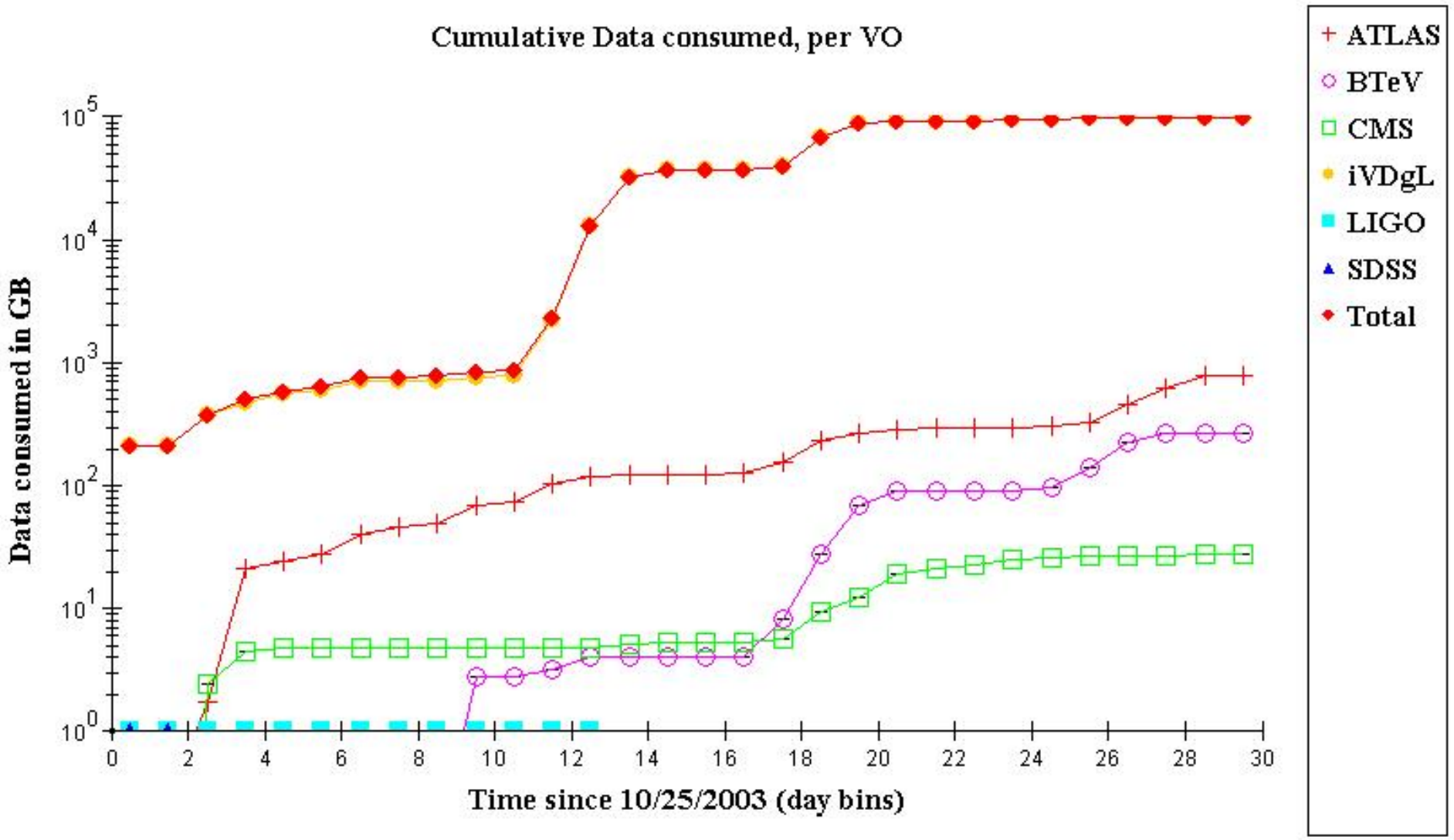
Grid3 – Cumulative CPU Days

to ~ 25 Nov 2003



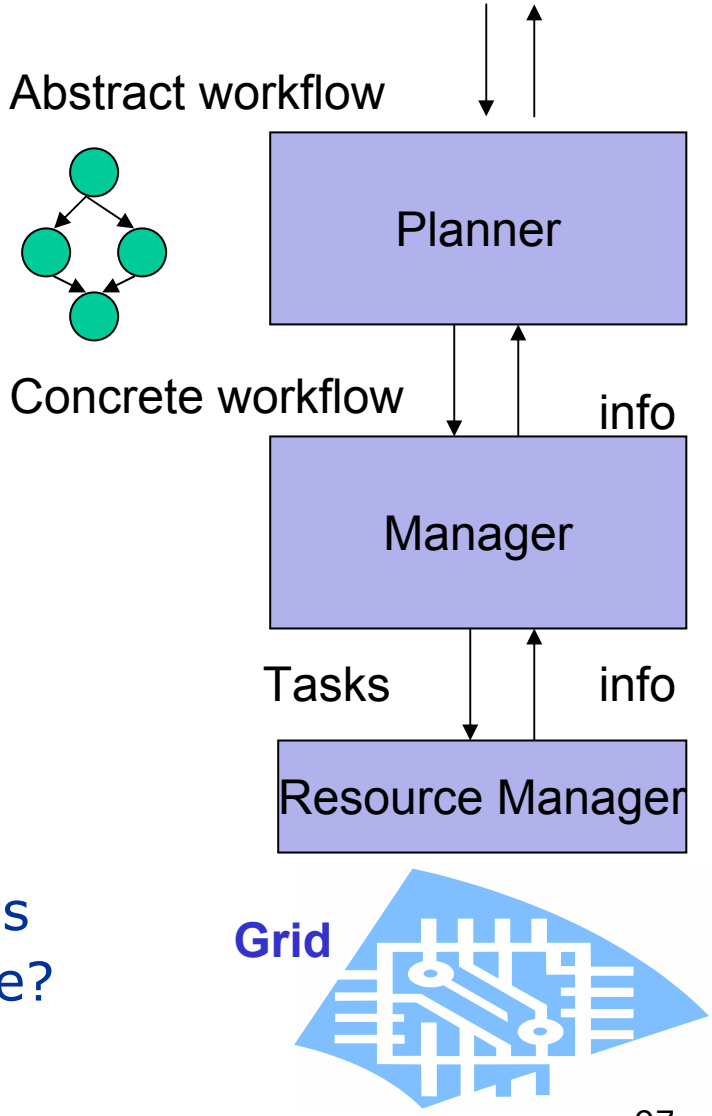
Grid2003: ~100TB data processed

to ~ 25 Nov 2003



Research issues

- Focus on data intensive science
- Planning is necessary
- Reaction to the environment is a must (things go wrong, resources come up)
- Iterative Workflow Execution:
  - Workflow Planner
  - Workload Manager
- Planning decision points
  - Workflow Delegation Time (eager)
  - Activity Scheduling Time (deferred)
  - Resource Availability Time (just in time)
- Decision specification level
- Reacting to the changing environment and recovering from failures
- How does the communication takes place? Callbacks, workflow annotations etc…



Future work

- Staging in executables on demand
- Expanding the scheduling plug-ins
- Investigating various partitioning approaches
- Investigating reliability across partitions

For further information

- Chimera and Pegasus:
  - www.griphyn.org/chimera
  - pegasus.isi.edu
- Workflow Management research group in GGF:
  - www.isi.edu/~deelman/wfm-rg

the globus alliance
www.globus.org

Outline

- Workflows on the Grid
- The GriPhyN project
- Chimera
- Pegasus
- Research issues
- Exercises