Workflows, Provenance and Virtual Data

Miron Livny
Computer Sciences Department
University of Wisconsin-Madison
miron@cs.wisc.edu
http://www.cs.wisc.edu/~miron

The Challenges of Data Placement and the need for Planning and Workflows

Customer requests:

Place $y = F(G(X))$ at L!

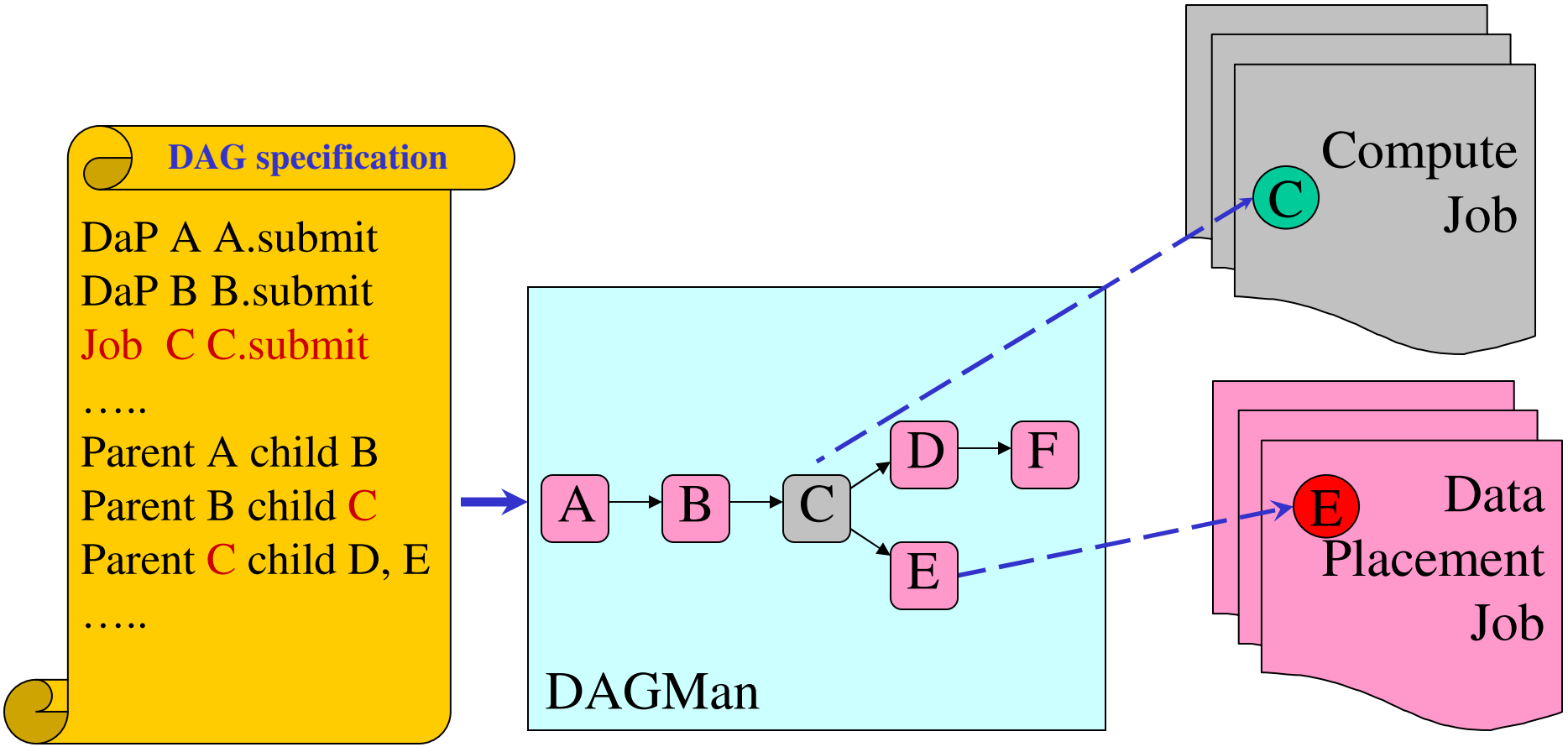
Master delivers.

Data Placement

Management of storage space and bulk data transfers play a key role in the end-to-end performance of an application.

- Data Placement (DaP) operations must be treated as "first class" jobs and explicitly expressed in the job flow
- Fabric must provide services to manage storage space
- Data Placement schedulers are needed.
- Data Placement and computing must be coordinated
- Smooth transition of CPU-I/O interleaving across software layers
- Error handling and garbage collection

Directed Acyclic Graphs (DAG)



A simple plan for y=F(x)→L

1. Allocate (size(x)+size(y)+size(F)) at SE(i)
2. Move x from SE(j) to SE(i)
3. Place F on CE(k)
4. Compute F(x) at CE(k)
5. Move y to L
6. Release allocated space

Storage Element (SE); Compute Element (CE)

Three levels of planning

Planning of the physical DAG

- All jobs at DAG submission time (eager),
- a job at a time when job is ready to go (lazy) or
- when resource is available (just in time)

Three layers of tools:

- Workflow specification - **Chimera**
- Planning – **Pegasus**
- Execution –

  DAGMan + Condor(G)

The GriPhyN Virtual Data System

GRIDS Center Community Workshop

Michael Wilde

wilde@mcs.anl.gov

Argonne National Laboratory

24 June 2005

Acknowledgements

…many thanks to the entire Trillium / OSG Collaboration, iVDGL and OSG Team, Virtual Data Toolkit Team, and all of our application science partners in ATLAS, CMS, LIGO, SDSS, Dartmouth DBIC and fMRIDC, SCEC, and Argonne's Computational Biology and Climate Science Groups of the Mathematics and Computer Science Division.

The Virtual Data System group is:

- ISI/USC: Ewa Deelman, Carl Kesselman, Gaurang Mehta, Gurmeet Singh, Mei-Hui Su, Karan Vahi
- U of Chicago: Catalin Dumitrescu, Ian Foster, Luiz Meyer (UFRJ, Brazil), Doug Scheftner, Jens Voeckler, Mike Wilde, Yong Zhao
- www.griphyn.org/vds

GriPhyN and iVDGL are supported by the National Science Foundation

Many of the research efforts involved in this work are supported by the US Department of Energy, office of Science.

The GriPhyN Project

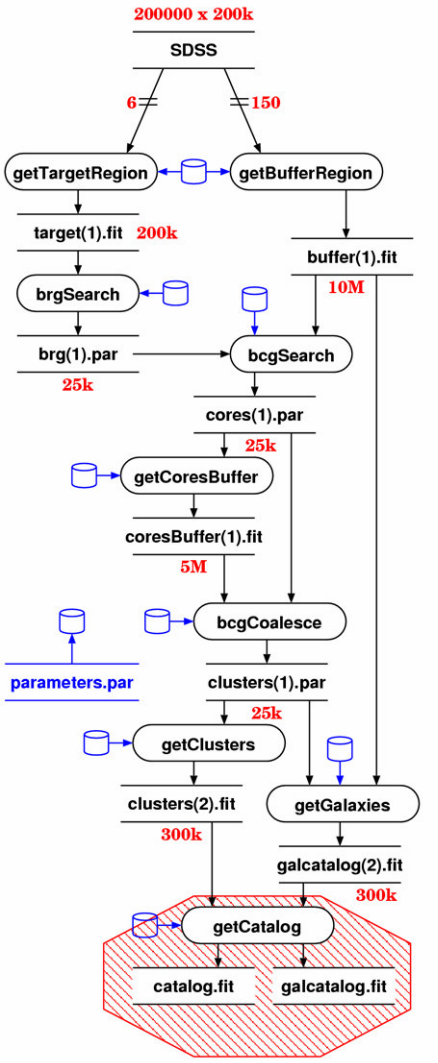
Enhance scientific productivity through…

- Discovery, application and management of data and processes at petabyte scale
- Using a worldwide data grid as a scientific workstation

The key to this approach is Virtual Data – creating and managing datasets through workflow "recipes" and provenance recording.

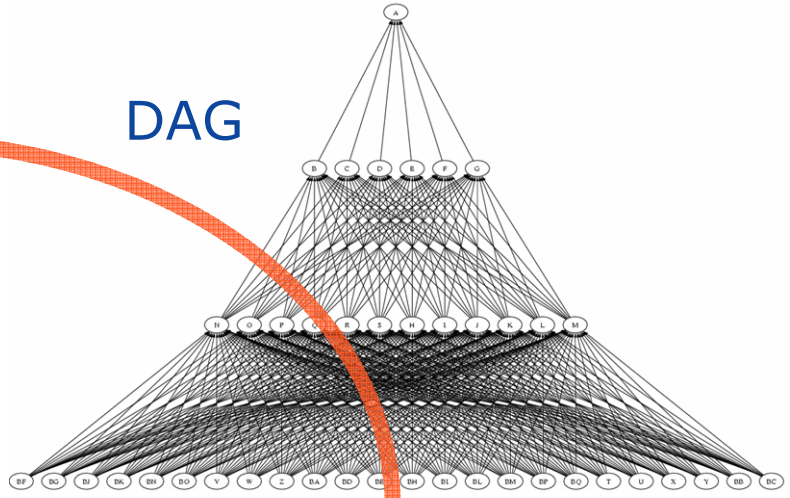
the globus alliance
www.globus.org

Virtual Data Example: *Galaxy Cluster Search*

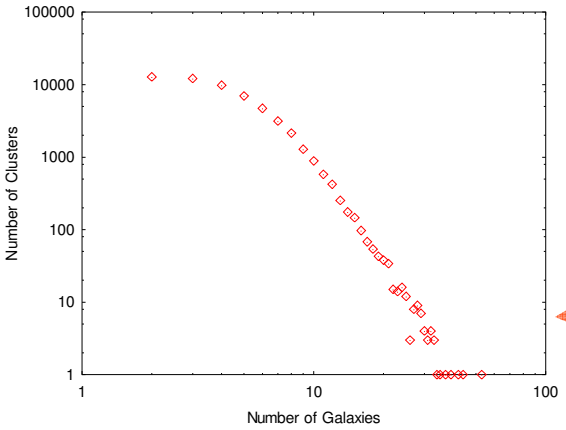


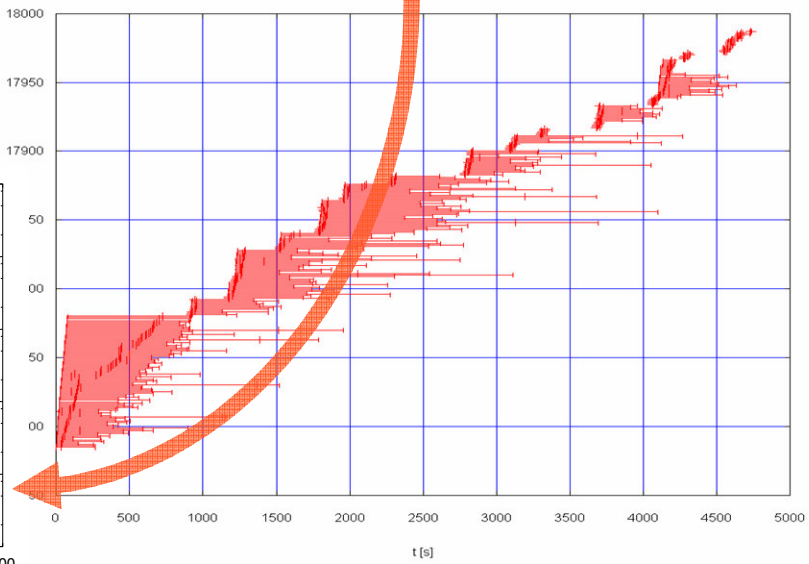
Sloan Data

DAG



Galaxy cluster size distribution





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

www.griphyn.org/

A virtual data glossary

- virtual data
  - defining data by the *logical* workflow needed to create it *virtualizes* it with respect to location, existence, failure, and representation
- VDS – Virtual Data System
  - The tools to define, store, manipulate and execute virtual data workflows
- VDT – Virtual Data Toolkit
  - A larger set of tools, based on NMI, VDT provides the Grid environment in which VDL workflows run
- VDL – Virtual Data Language
  - A language (text and XML) that defines the functions and function calls of a virtual data workflow
- VDC – Virtual Data Catalog
  - The database and schema that store VDL definitions

What must we "virtualize" to compute on the Grid?

- Location-independent computing: represent all workflow in abstract terms
- Declarations not tied to specific entities:
  - sites
  - file systems
  - schedulers
- Failures – automated retry for data server and execution site un-availability

Expressing Workflow in VDL

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

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

DV grep (a1=@{in:file1}, a2=@{out:file2});
DV sort (a1=@{in:file2}, a2=@{out:file3});
```

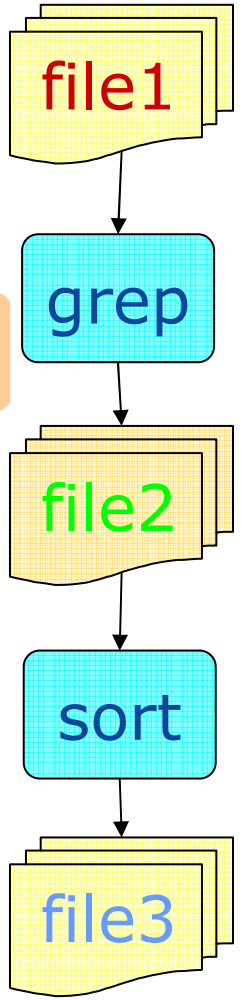
Define a "function" wrapper for an application

Define "formal arguments" for the application

Connect applications via output-to-input dependencies

Define a "call" to invoke application

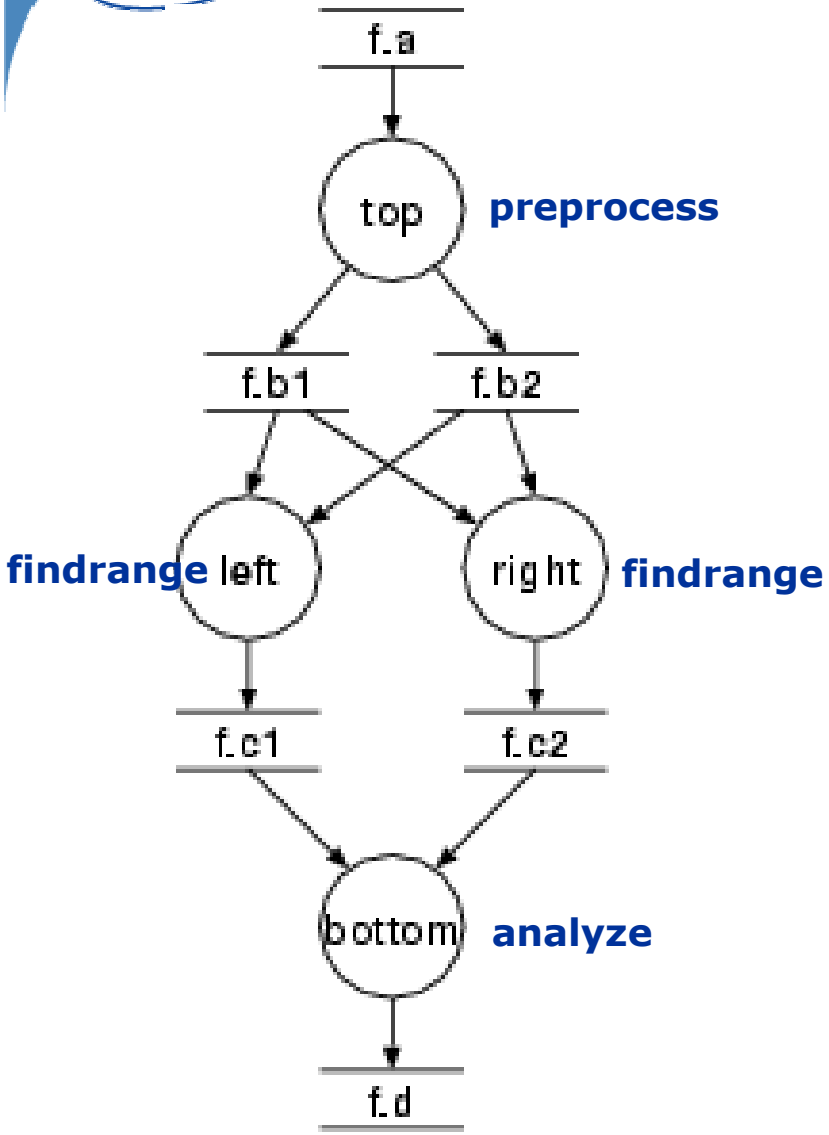
Provide "actual" argument values for the invocation



Essence of VDL

- Elevates specification of computation to a logical, location-independent level
- Acts as an "interface definition language" at the shell/application level
- Can express composition of functions
- Codable in textual and XML form
- Often machine-generated to provide ease of use and higher-level features
- Preprocessor provides iteration and variables

Compound Workflow



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

Compound Transformations for nesting Workflows

- Compound TR encapsulates an entire sub-graph:

```
TR rangeAnalysis (in fa, p1, p2,
                  out fd, io fc1,
                  io fc2, io fb1, io fb2, )
{
 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-> rangeAnalysis ( 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-> rangeAnalysis ( 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-> rangeAnalysis ( 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" );
```

the globus alliance
www.globus.org

Using VDL

GriPhyN

- Generated directly for low-volume usage
- Generated by scripts for production use
- Generated by application tool builders as wrappers around scripts provided for community use
- Generated transparently in an application-specific portal (e.g. quarknet.fnal.gov/grid)
- Generated by drag-and-drop workflow design tools such as Triana

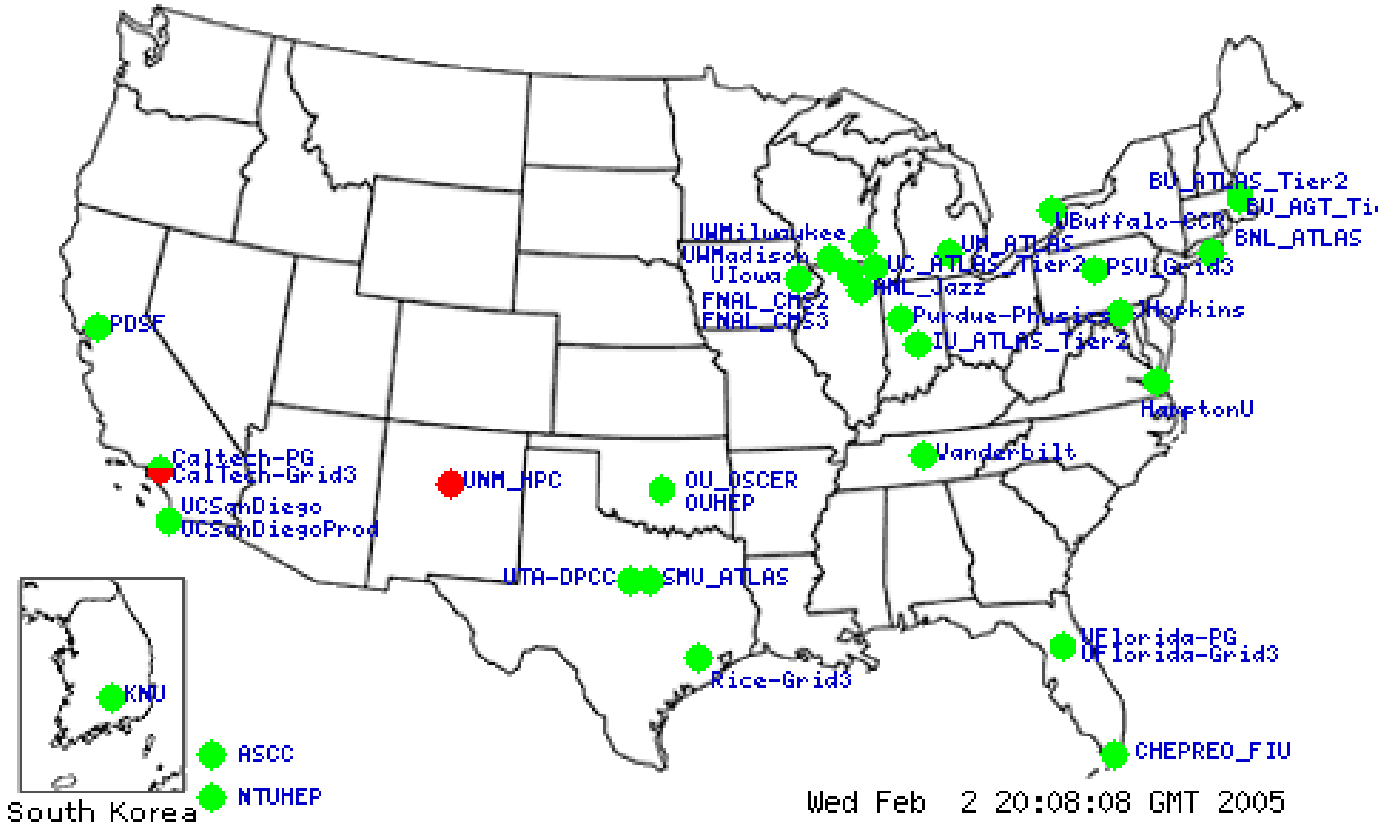
Basic VDL Toolkit

- Convert between text and XML representation
- Insert, update, remove definitions from a virtual data catalog
- Attach metadata annotations to defintions
- Search for definitions
- Generate an abstract workflow for a data derivation request
- Multiple interface levels provided:
  - Java API, command line, web service

Representing Workflow

- Specifies a set of activities and control flow
- Sequences information transfer between activities
- VDS uses XML-based notation called "DAG in XML" (DAX) format
- VDC Represents a wide range of workflow possibilities
- DAX document represents steps to create a specific data product

OSG:The "target chip" for VDS Workflows



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

www.griphyn.org/vds

VDS Applications

Application	Jobs / workflow	Levels	Status
ATLAS HEP Event Simulation	500K	1	In Use
LIGO Inspiral/Pulsar	~700	2-5	Inspiral In Use
NVO/NASA Montage/Morphology	1000s	7	Both In Use
GADU Genomics: BLAST,…	40K	1	In Use
fMRI DBIC AIRSN Image Proc	100s	12	In Devel
QuarkNet CosmicRay science	<10	3-6	In Use
SDSS Coadd; Cluster Search	40K 500K	2 8	In Devel / CS Research
FOAM Ocean/Atmos Model	2000 (core app runs 250 8-CPU jobs)	3	In use
GTOMO Image proc	1000s	1	In Devel
SCEC Earthquake sim	1000s		In use

A Case Study – Functional MRI

- Problem: "spatial normalization" of a images to prepare data from fMRI studies for analysis
- Target community is approximately 60 users at Dartmouth Brain Imaging Center
- Wish to share data and methods across country with researchers at Berkeley
- Process data from arbitrary user and archival directories in the center's AFS space; bring data back to same directories
- Grid needs to be transparent to the users: Literally, "Grid as a Workstation"

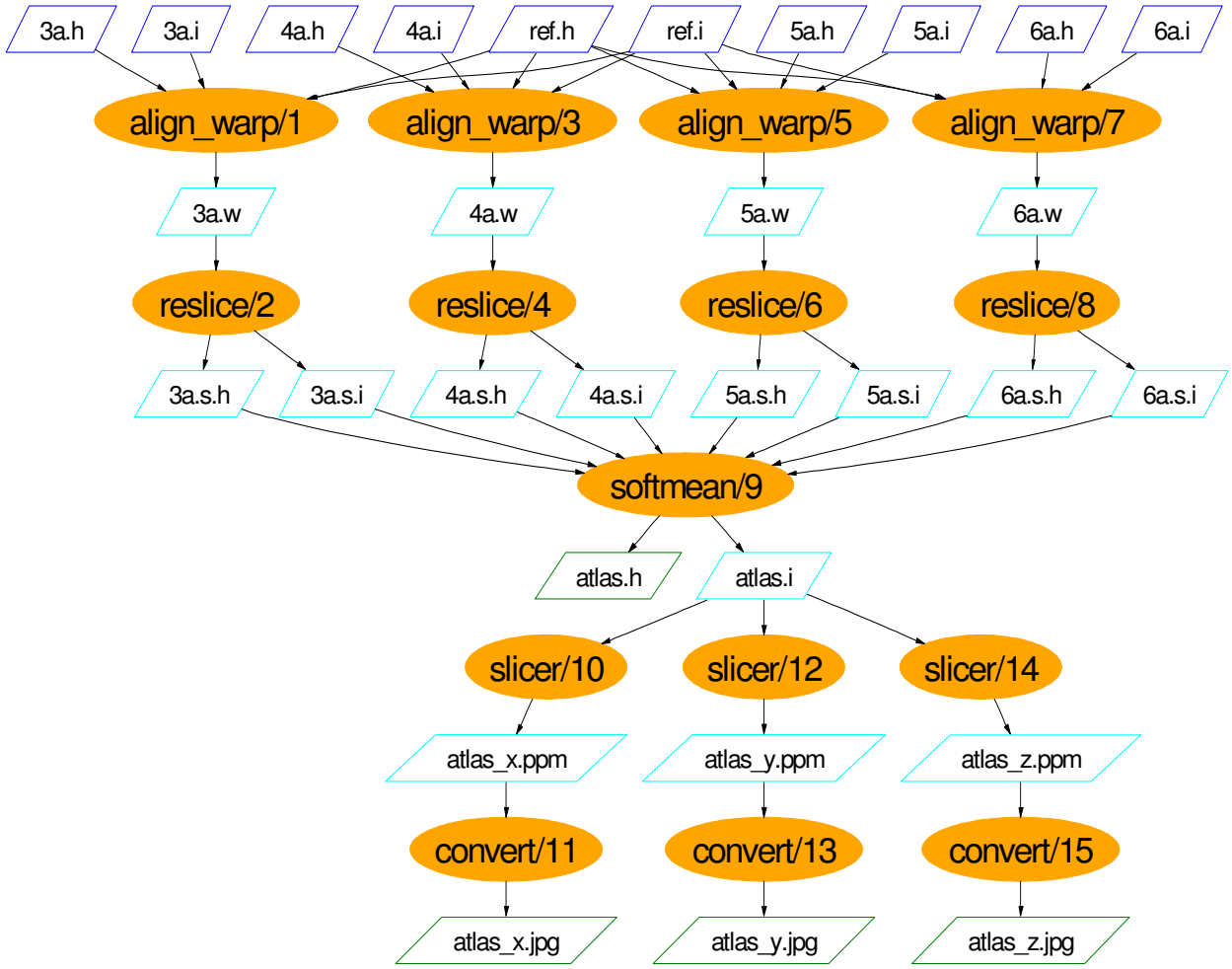
A Case Study – Functional MRI (2)

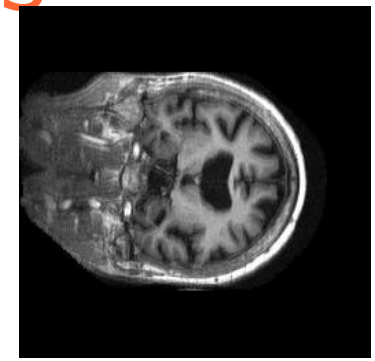
- Based workflow on shell script that performs 12-stage process on a local workstation
- Adopted replica naming convention for moving user's data to Grid sites
- Creates VDL pre-processor to iterate transformations over datasets
- Utilizing resources across two distinct grids – Grid3 and Dartmouth Green Grid

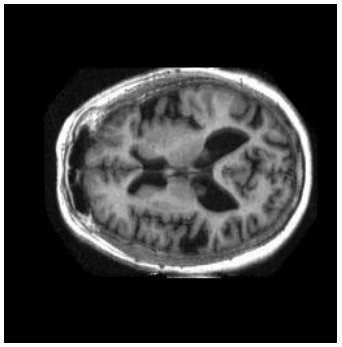
the globus alliance
www.globus.org

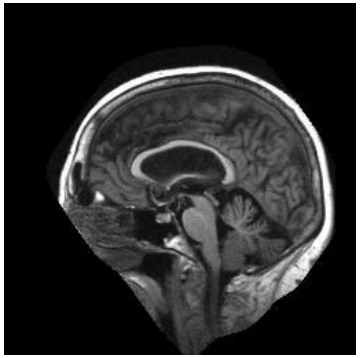
GriPhyN

Functional MRI Analysis









Workflow courtesy James Dobson, Dartmouth Brain Imaging Center

www.griphyn.org/vds

fMRI Dataset processing


```
FOREACH BOLDSEQ
DV reorient (# Process Blood O2 Level Dependent Sequence
  input =   [  @{in:  "$BOLDSEQ.img"},
               @{in:  "$BOLDSEQ.hdr"}   ],
  output = [@{out: "$CWD/FUNCTIONAL/r$BOLDSEQ.img"}
            @{out: "$CWD/FUNCTIONAL/r$BOLDSEQ.hdr"}],
  direction = "y", );
END

DV softmean (
  input = [ FOREACH BOLDSEQ
      @{in:"$CWD/FUNCTIONAL/har$BOLDSEQ.img"}
  END ],
  mean = [ @{out:"$CWD/FUNCTIONAL/mean"} ]
);
```

the globus alliance
www.globus.org

GriPhyN

fMRI Virtual Data Queries

Which transformations can process a "subject image"?

- Q: xsearchvdc -q tr_meta dataType subject_image input
- A: fMRIDC.AIR::align_warp

List anonymized subject-images for young subjects:

- Q: xsearchvdc -q lfn_meta dataType subject_image privacy anonymized subjectType young
- A: 3472-4_anonymized.img

Show files that were derived from patient image 3472-3:

- Q: xsearchvdc -q lfn_tree 3472-3_anonymized.img
- A: 3472-3_anonymized.img
  3472-3_anonymized.sliced.hdr
  atlas.hdr
  atlas.img
  …
  atlas_z.jpg
  3472-3_anonymized.sliced.img

Provenance for ATLAS DC2

How much compute time was delivered?

years	mon	year
.45	6	2004
20	7	2004
34	8	2004
40	9	2004
15	10	2004
15	11	2004
8.9	12	2004

Selected statistics for one of these jobs:

```
start: 2004-09-30 18:33:56
duration: 76103.33
     pid: 6123
exitcode: 0
    args: 8.0.5 JobTransforms-08-00-05-09/share/dc2.g4sim.filter.trf CPE_6785_556
          ... -6 6 2000 4000 8923 dc2_B4_filter_frag.txt
   utime: 75335.86
   stime: 28.88
  minflt: 862341
  majflt: 96386
```

Which Linux kernel releases were used ?

How many jobs were run on a Linux 2.4.28 Kernel?

Conclusion

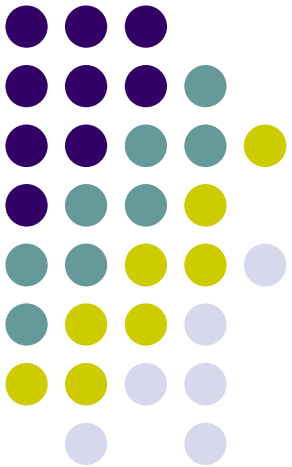
- Using VDL to express location-independent computing is proving effective: science users save time by using it over ad-hoc methods
  - VDL automates many complex and tedious aspects of distributed computing
- Proving capable of expressing workflows across numerous sciences and diverse data models: HEP, Genomics, Astronomy, Biomedical
- Makes possible new capabilities and methods for data-intensive science based on its uniform provenance model
- Provides an abstract front-end for Condor workflow, automating DAG creation

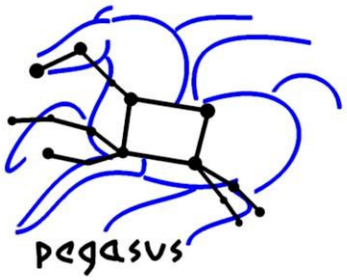
Next Steps

- Unified representation of data-sets, metadata, provenance and mappings to physical storage
- Improved queries to discover existing products and to perform incremental work (versioning)
- Improved error handling and diagnosis: VDS is like a compiler whose target chip architecture is the Grid – this is a tall order, and much work remains.
- Leverage XQuery to formulate new workflows from those in a VO's catalogs

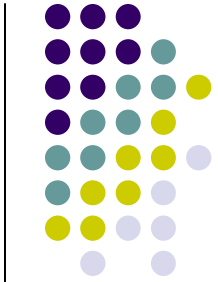
Large-Scale Science Through Workflow Management

Ewa Deelman
Center for Grid Technologies
USC Information Sciences Institute

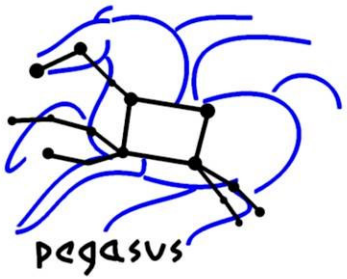




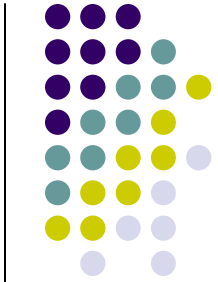
Acknowledgements



- Ewa Deelman, Carl Kesselman, Gaurang Mehta, Gurmeet Singh, Mei-Hui Su, Karan Vahi (Center for Grid Technologies, 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 and EU-funded GridLab



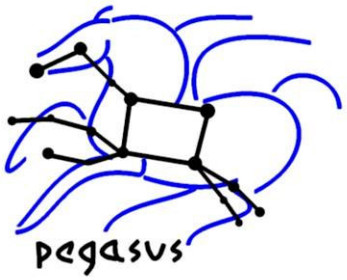
Today's Scientific Applications



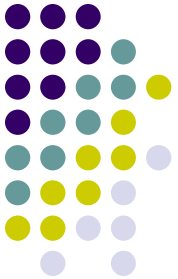
- Increasing in the level of complexity
- Use of individual application components
- Reuse of individual intermediate data products (files)
- Description of Data Products using Metadata Attributes

- 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



Workflow Definitions

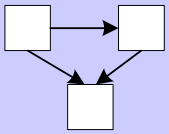


- Workflow template: shows the main steps in the scientific analysis and their dependencies without specifying particular data products
- Abstract workflow: depicts the scientific analysis including the data used and generated, but does not include information about the resources needed for execution
- Concrete workflow: an executable workflow that includes details of the execution environment

Scientific Analysis

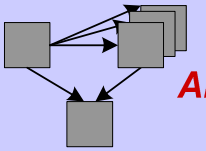
Workflow Evolution

Construct the Analysis



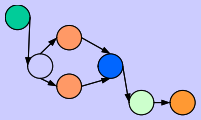
Workflow Template

Select the Input Data



Abstract Worfklow

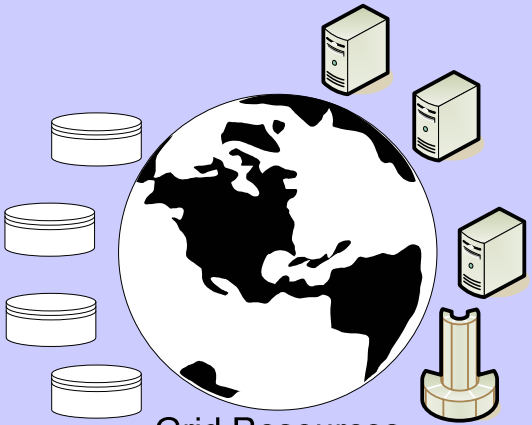
Map the Workflow onto Available Resources



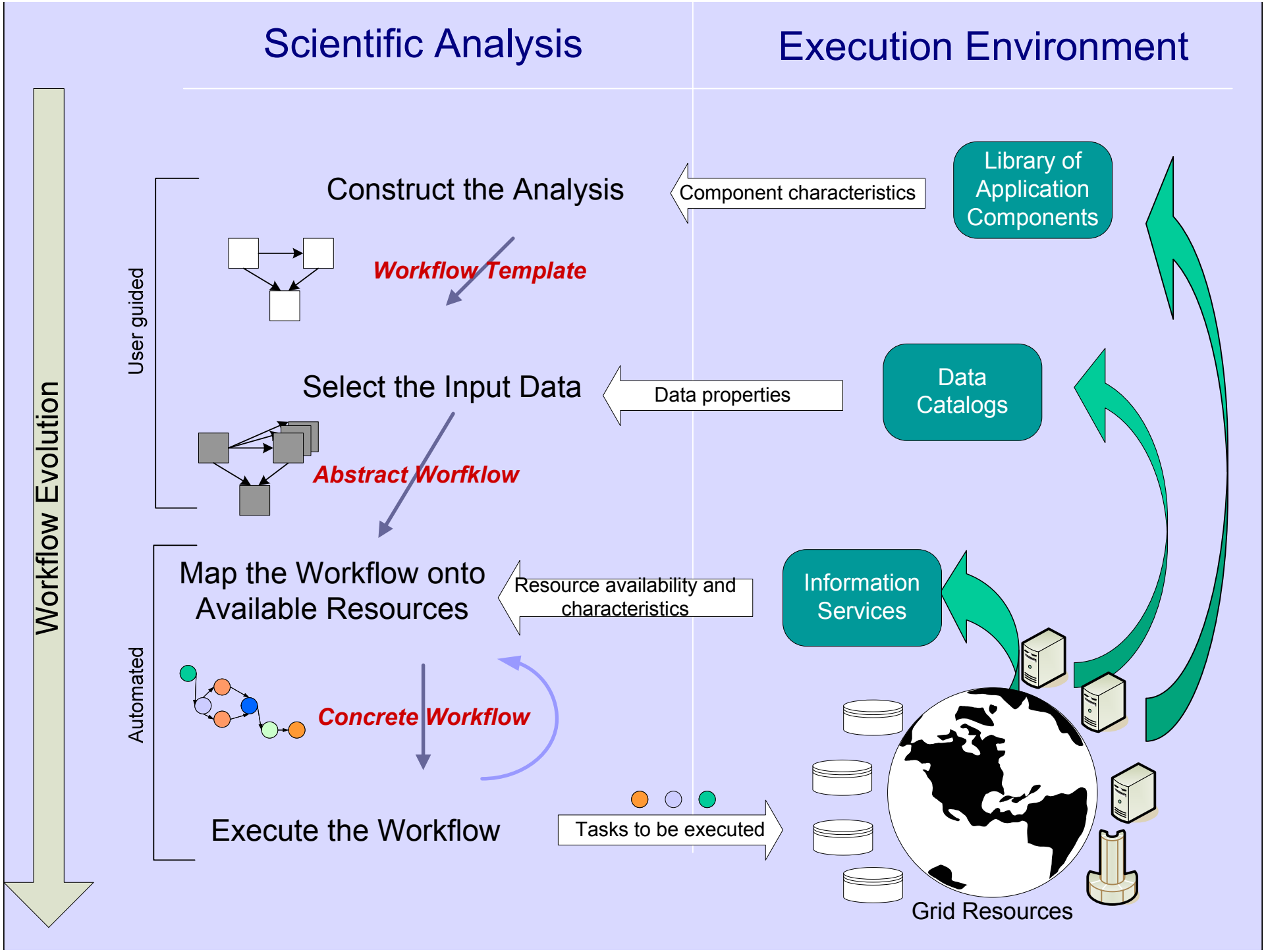
Concrete Workflow

Execute the Workflow

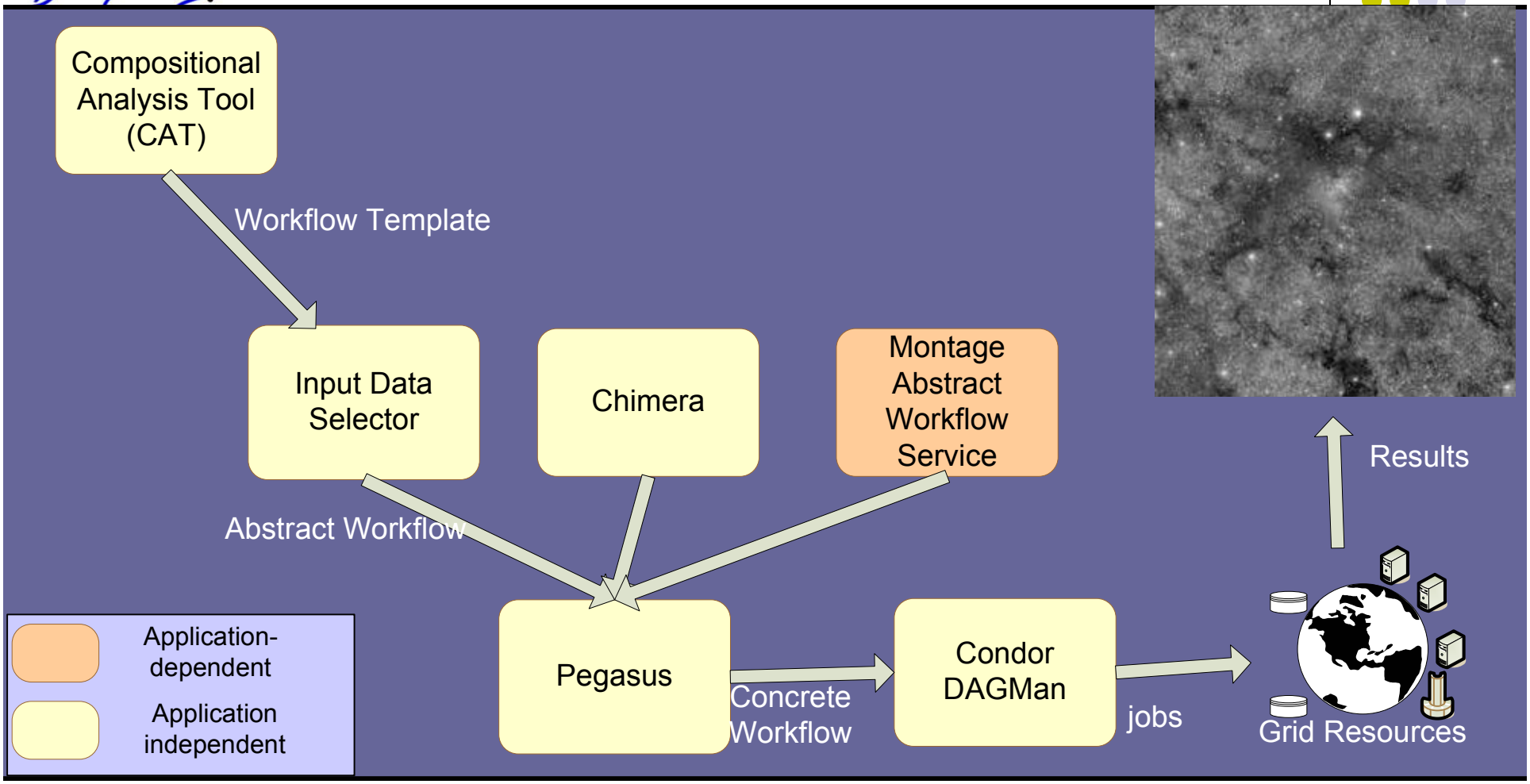
Tasks to be executed

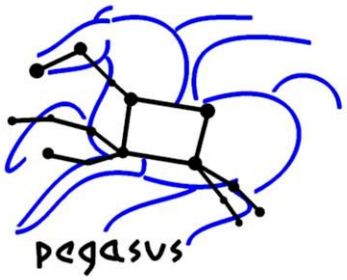


Grid Resources

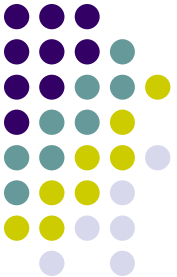

Scientific Analysis
Execution Environment
Workflow Evolution
User guided
Automated
Construct the Analysis
Workflow Template
Select the Input Data
Abstract Worfklow
Map the Workflow onto Available Resources
Concrete Workflow
Execute the Workflow
Component characteristics
Library of Application Components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

Concrete Workflow Generation and Mapping

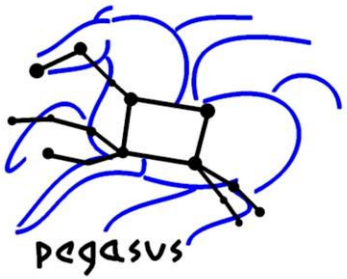




Pegasus: Planning for Execution in Grids



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



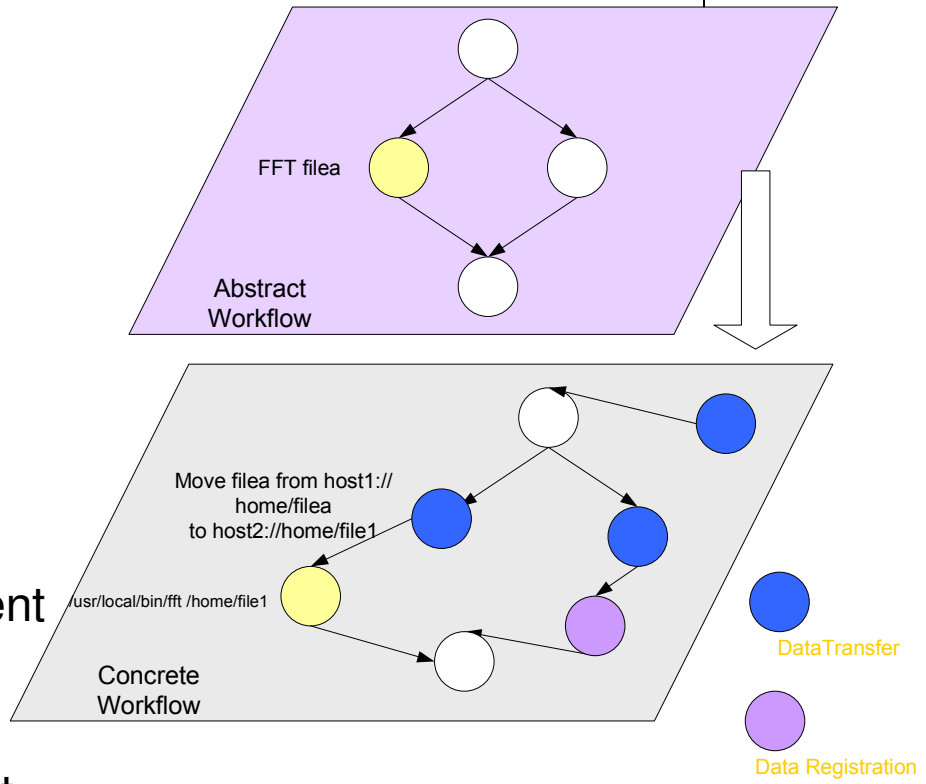
Generating a Concrete Workflow

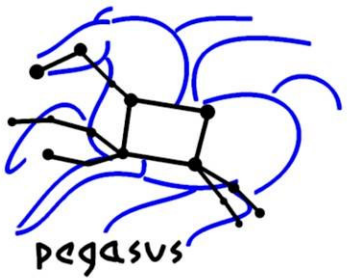
Information

- location of files and component Instances
- State of the Grid resources

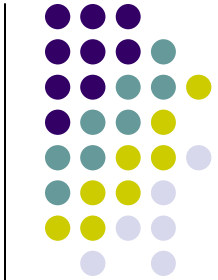
Select specific

- 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

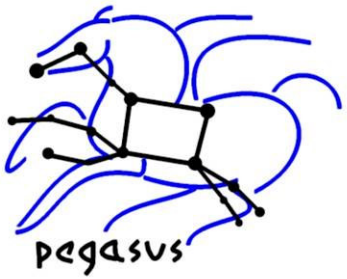




Information Components used by Pegasus

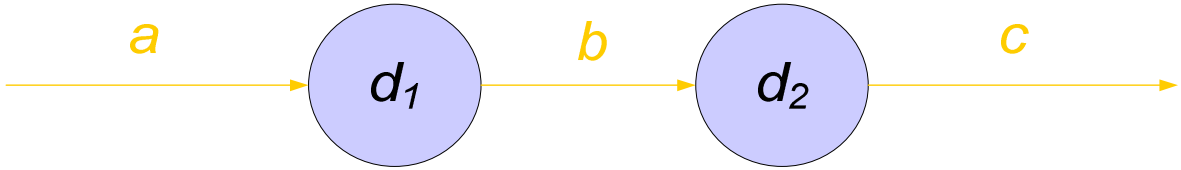


- Globus Monitoring and Discovery Service (MDS)
  - Locates available resources
  - Finds resource properties
    - Dynamic: load, queue length
    - Static: location of GridFTP server, RLS, etc
- Globus Replica Location Service
  - Locates data that may be replicated
  - Registers new data products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

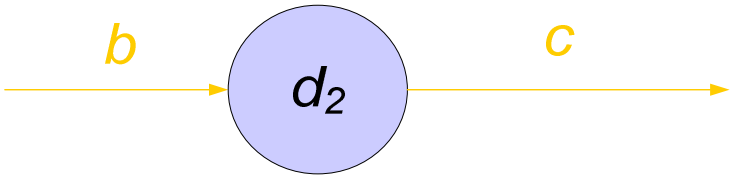


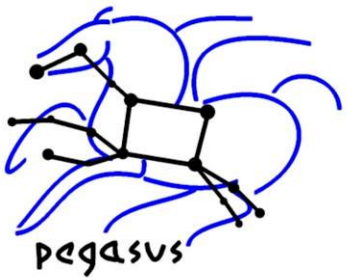
Example Workflow Reduction

- Original abstract workflow

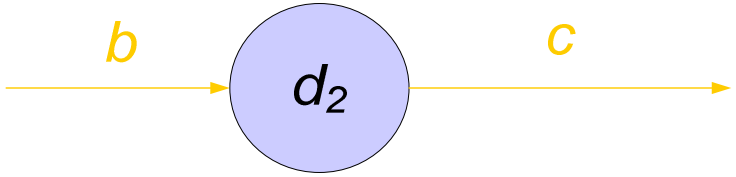


- If “b” already exists (as determined by query to the RLS), the workflow can be reduced

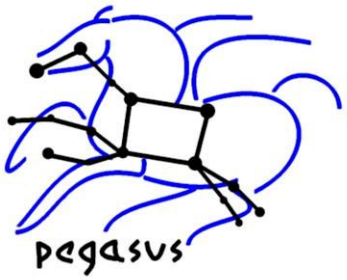




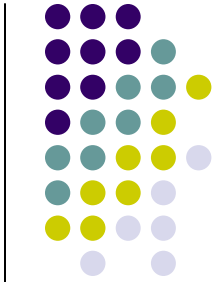
Mapping from abstract to concrete



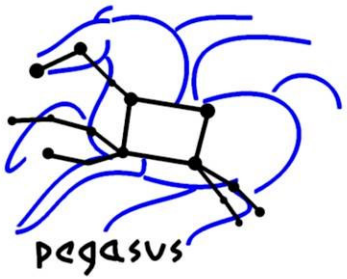
- Query RLS, MDS, and TC, schedule computation and data movement



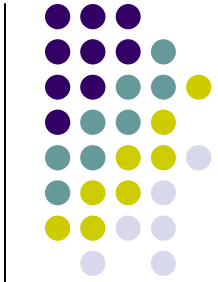
Pegasus Research



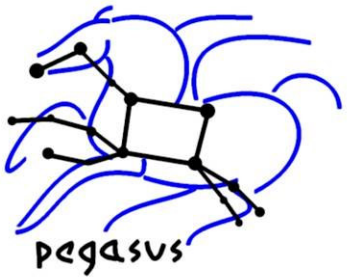
- resource discovery and assessment
- resource selection
- resource provisioning
- workflow restructuring
  - task merged together or reordered to improve overall performance
- adaptive computing
  - Workflow refinement adapts to changing execution environment



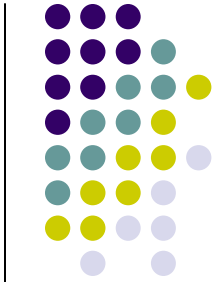
Benefits of the workflow & Pegasus approach



- The workflow exposes
  - the structure of the application
  - maximum parallelism of the application
- Pegasus can take advantage of the structure to
  - Set a planning horizon (how far into the workflow to plan)
  - Cluster a set of workflow nodes to be executed as one (for performance)
- Pegasus shields from the Grid details



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.

Mosaic of M42 created on the Teragrid resources using Pegasus

Pegasus improved the runtime of this application by 90% over the baseline case

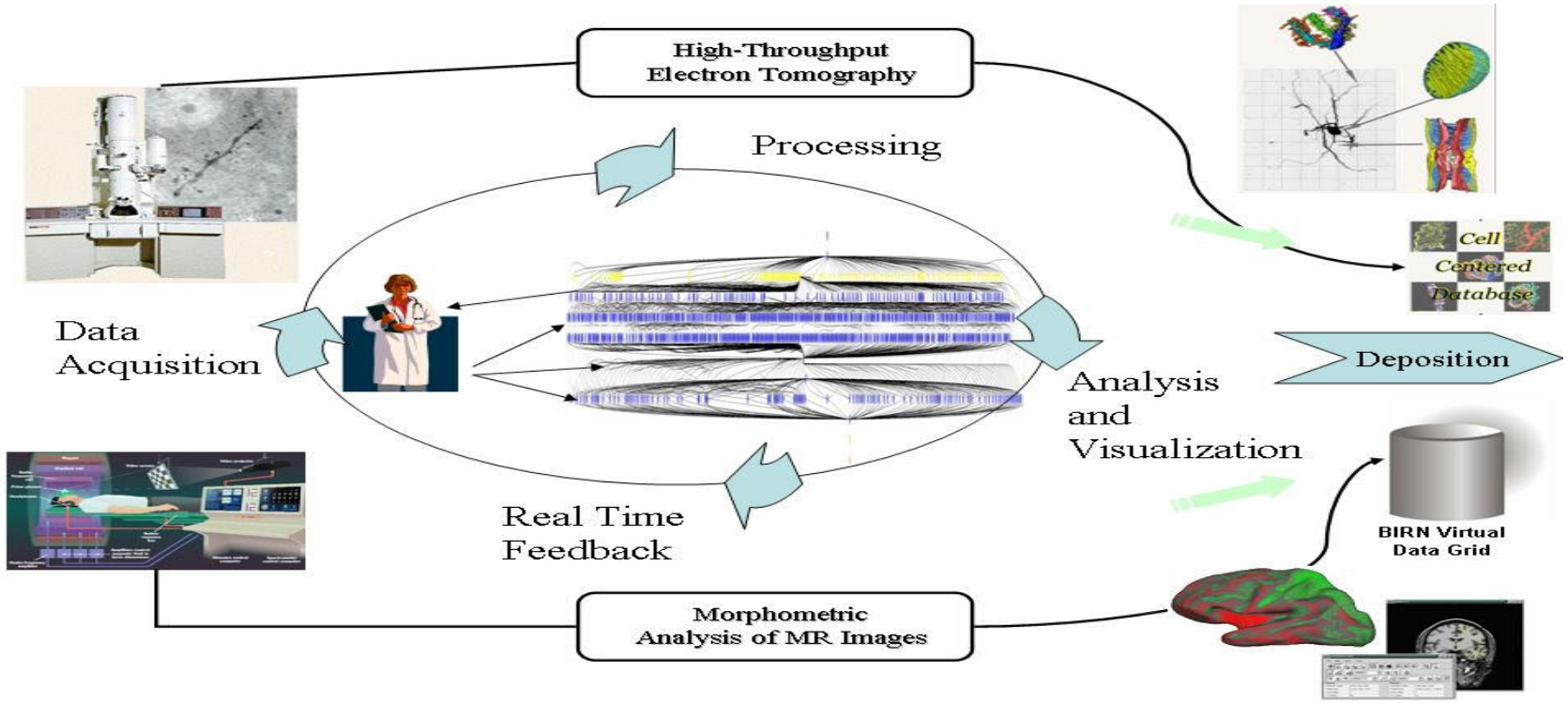
Bruce Berriman, John Good (Caltech)

Joe Jacob, Dan Katz (JPL)

Future Directions

Support for workflows with real-time feedback to scientists.

Providing intermediate analysis results so that the experimental setup can be adjusted while the short-lived samples or human subjects are available.

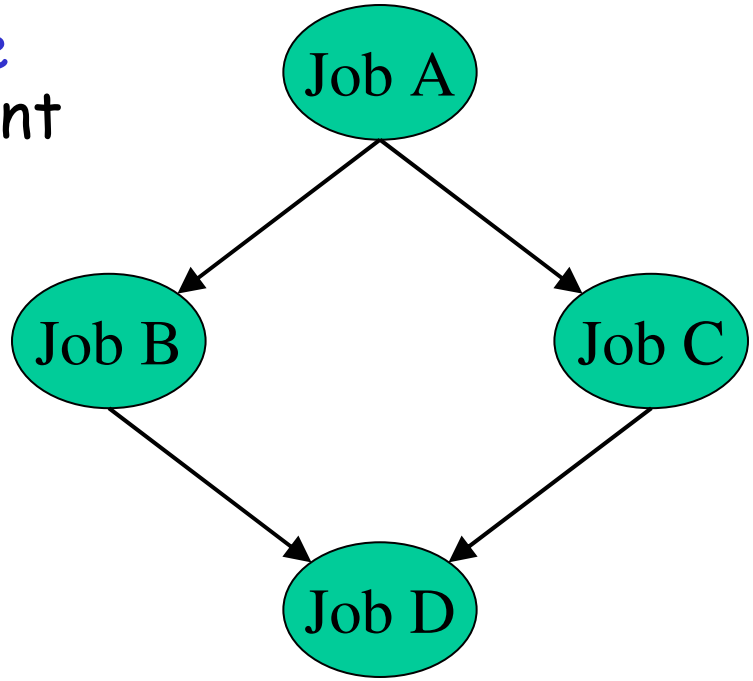


DAGMan

- **Directed Acyclic Graph Manager**
- DAGMan allows you to specify the *dependencies* between your Condor-G jobs, so it can *manage* them automatically for you.
- (e.g., "Don't run job "B" until job "A" has completed successfully.")

What is a DAG?

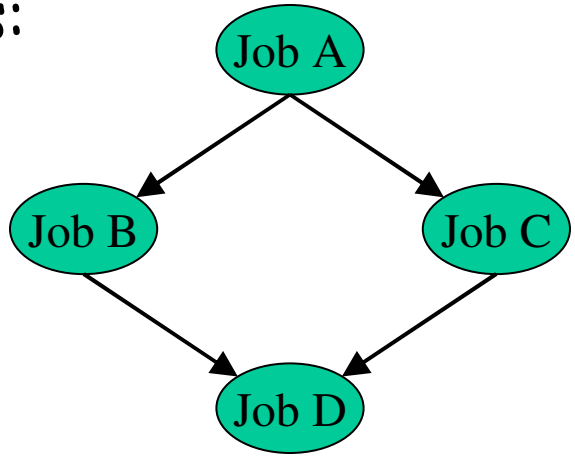
- A DAG is the data structure used by DAGMan to represent these dependencies.
- Each job is a "node" in the DAG.
- Each node can have any number of "parent" or "children" nodes – as long as there are no loops!



Defining a DAG

- A DAG is defined by a *.dag file*, listing each of its nodes and their dependencies:

```
# diamond.dag
Job A a.sub
Job B b.sub
Job C c.sub
Job D d.sub
Parent A Child B C
Parent B C Child D
```

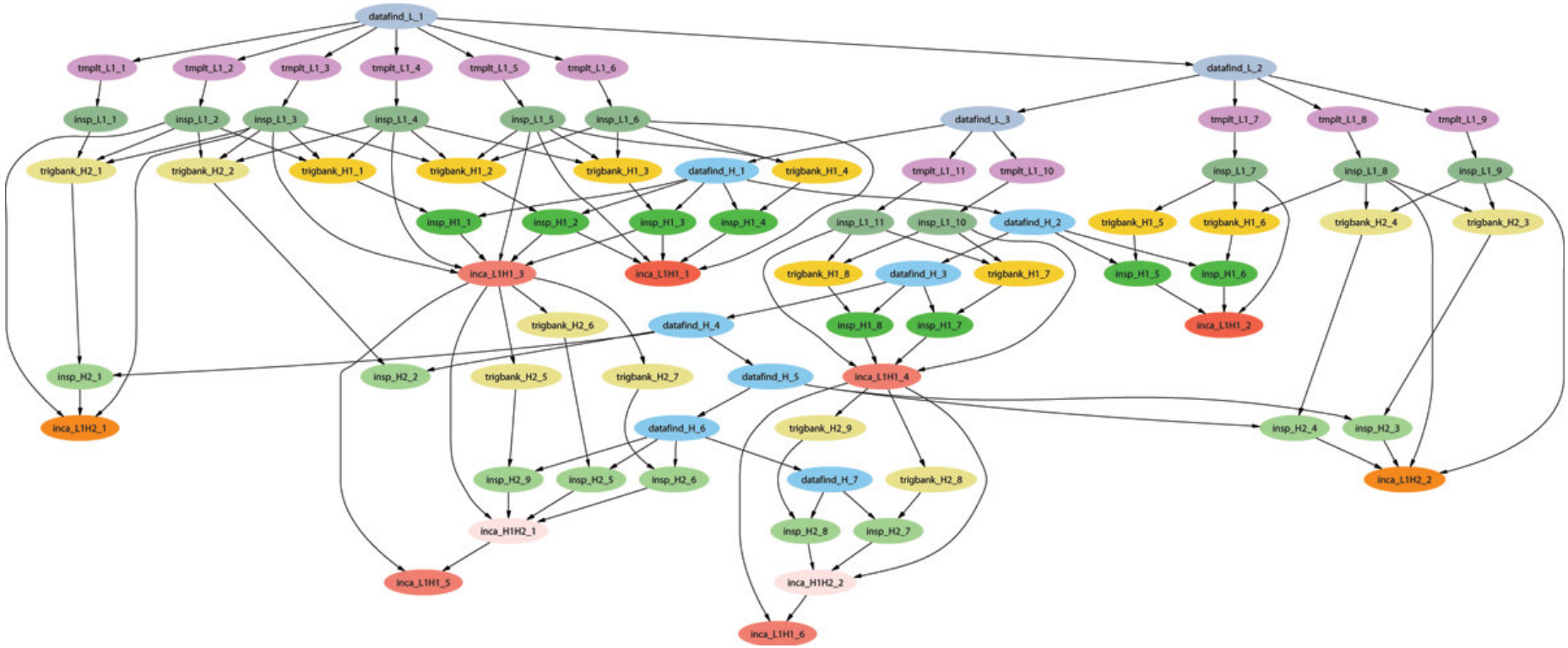


- each node will run the Condor-G job specified by its accompanying Condor submit file

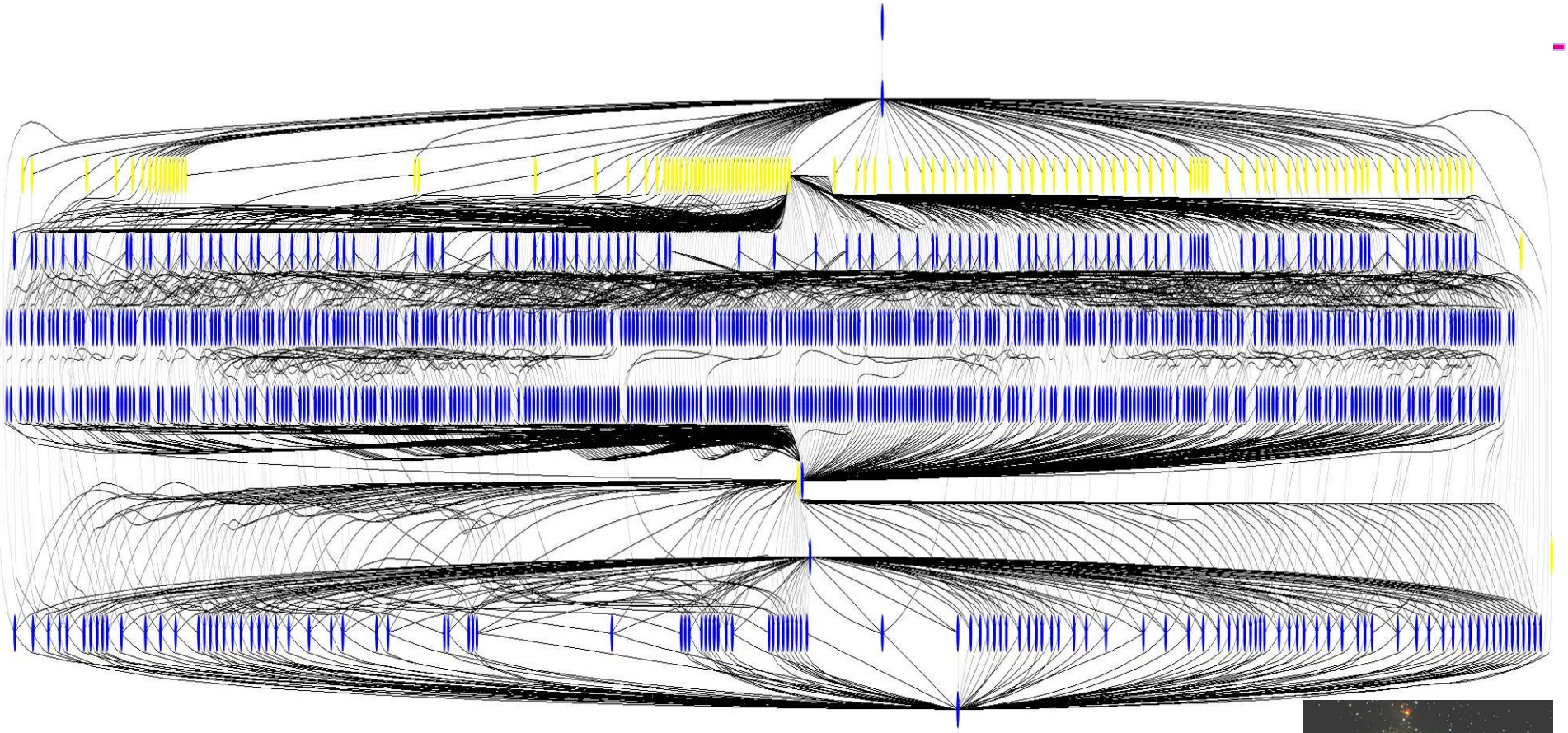
DAGMan characteristics

- Lightweight and portable approach to workflow management
- Enables recursive DAGs
- Supports distributed workflows
- Can be modified/enhanced by user to reflect personal/local policies
- Dependable and flexible
- The "work horse" of most grids

Example of a LIGO Inspiral DAG

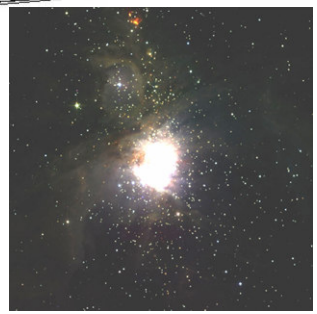


Small Montage Workflow

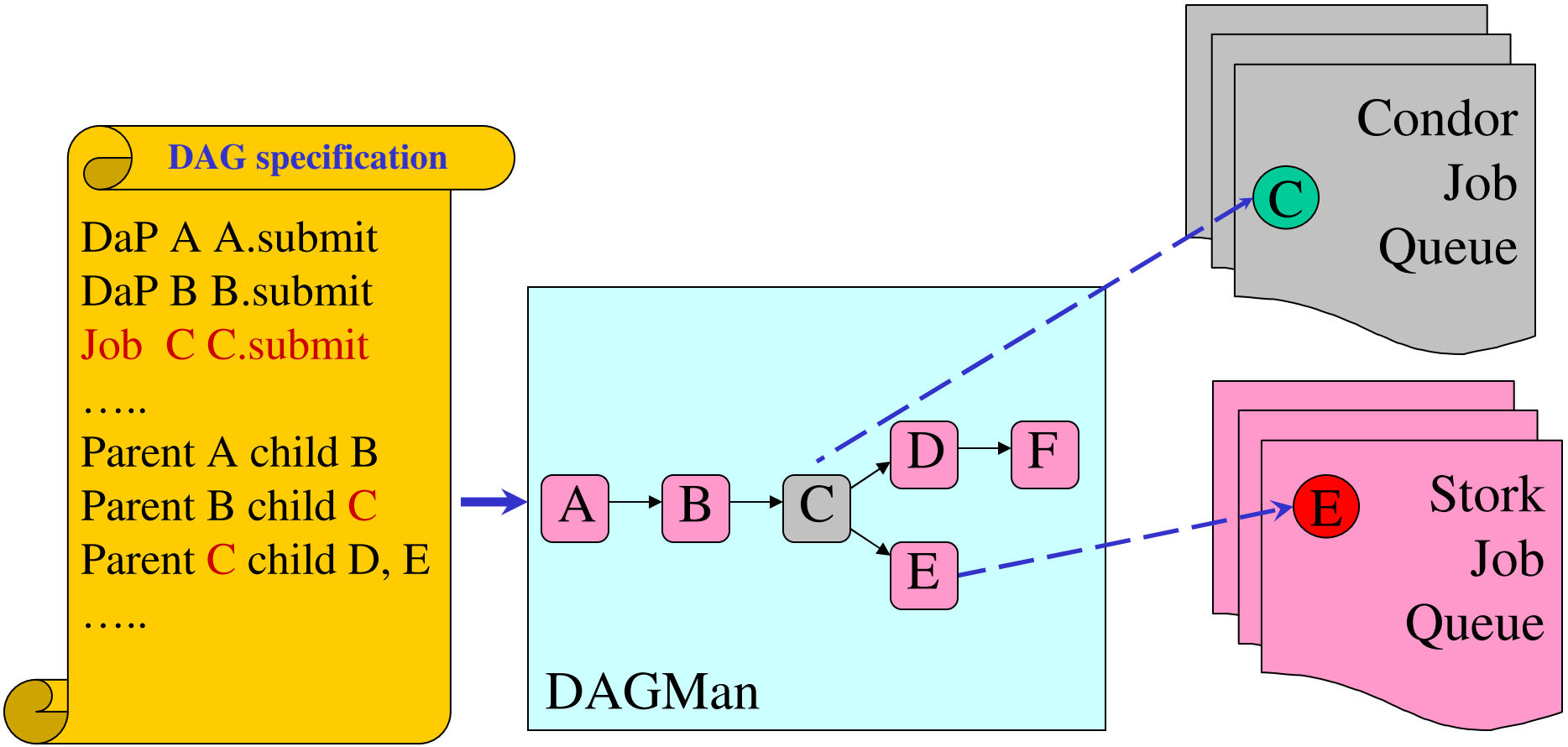


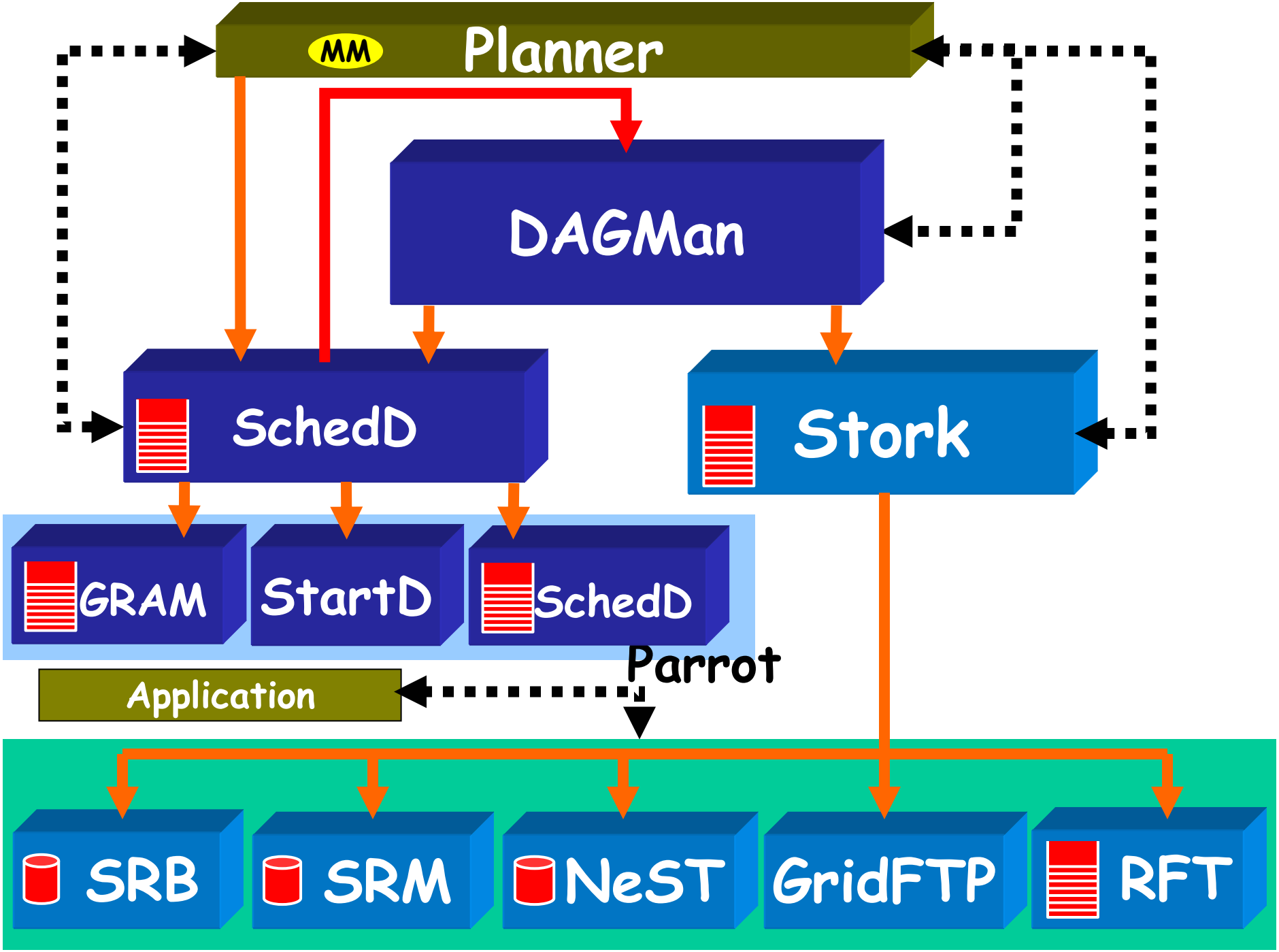
~1200 node workflow, 7 levels

Mosaic of M42 created on the Teragrid using Pegasus



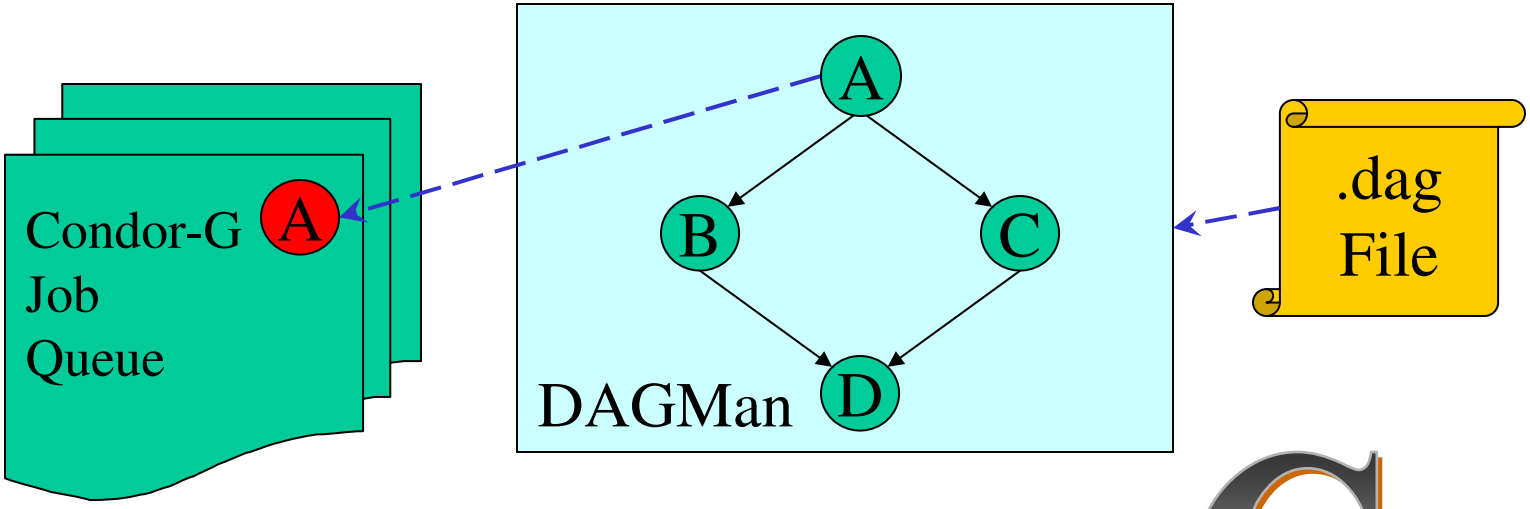
Directed Acyclic Graphs (DAG)




MM
Planner
DAGMan
SchedD
Stork
GRAM
StartD
SchedD
Parrot
Application
SRB
SRM
NeST
GridFTP
RFT

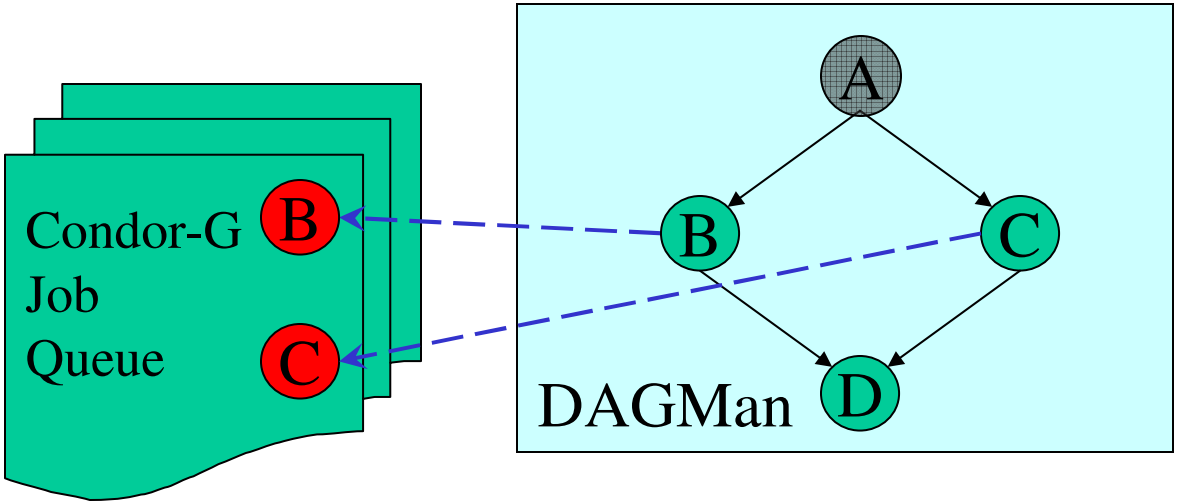
Running a DAG

- DAGMan acts as a "meta-scheduler", managing the submission of your jobs to Condor-G based on the DAG dependencies.



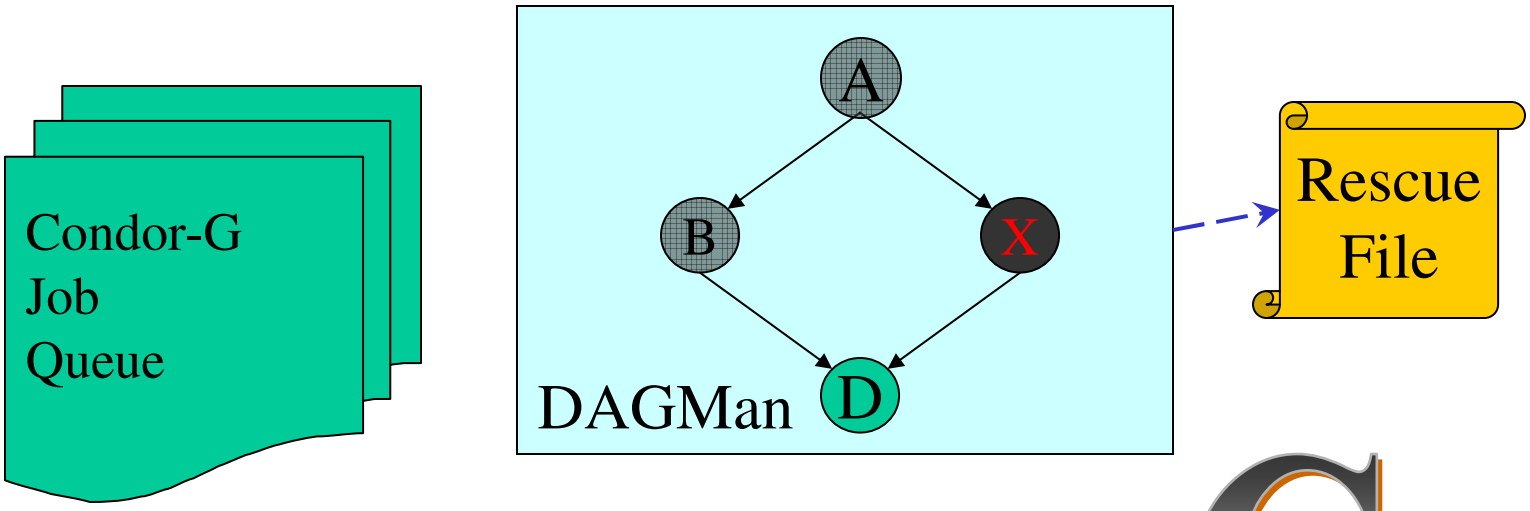
Running a DAG (cont'd)

- DAGMan holds & submits jobs to the Condor-G queue at the appropriate times.



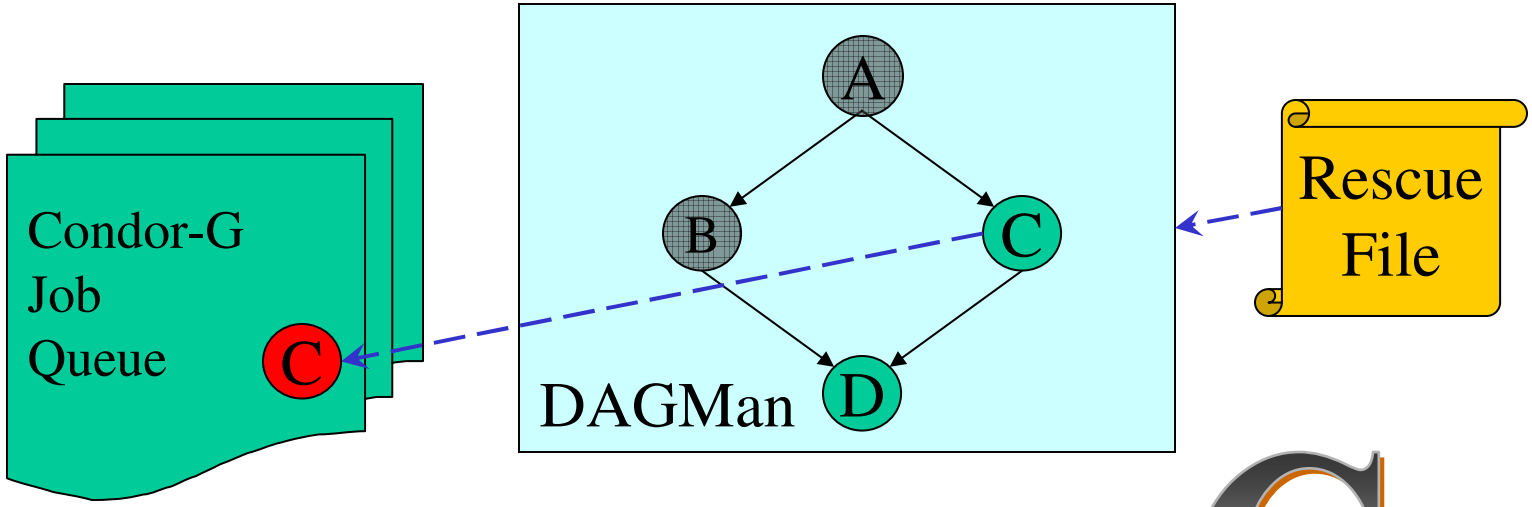
Running a DAG (cont'd)

- In case of a job failure, DAGMan continues until it can no longer make progress, and then creates a *"rescue" file* with the current state of the DAG.



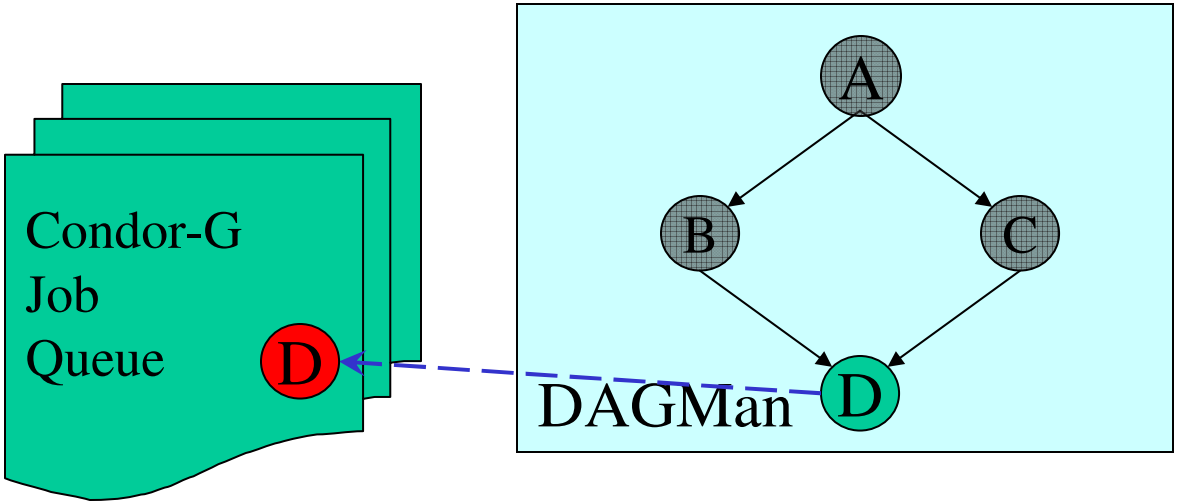
Recovering a DAG

- Once the failed job is ready to be re-run, the rescue file can be used to restore the prior state of the DAG.



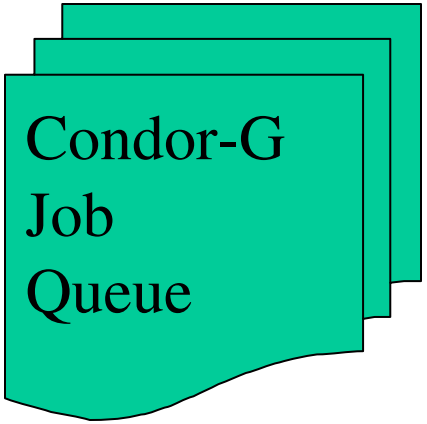
Recovering a DAG (cont'd)

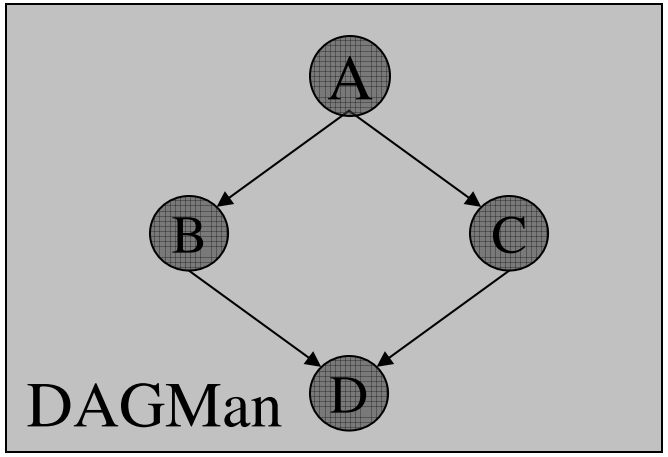
- Once that job completes, DAGMan will continue the DAG as if the failure never happened.



Finishing a DAG

- Once the DAG is complete, the DAGMan job itself is finished, and exits.





Additional DAGMan Features

- Provides other handy features for job management…
  - nodes can have PRE & POST scripts
  - failed nodes can be automatically re-tried a configurable number of times

Real life Data Pipelines

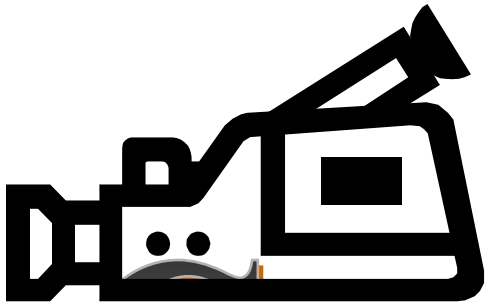
NCSA

ASTRONOMY
UNIVERSITY OF ILLINOIS AT URBANA–CHAMPAIGN

- Astronomy data processing pipeline
  - ~3 TB (2611 x 1.1 GB files)
  - Joint work with Robert Brunner, Michael Remijan et al. at NCSA

WCER

Wisconsin Center for Education Research
at the School of Education, University of Wisconsin-Madison



- WCER educational video pipeline
  - ~6TB (13 GB files)
  - Joint work with Chris Thorn et al at WCER

Digital Insight
WCER

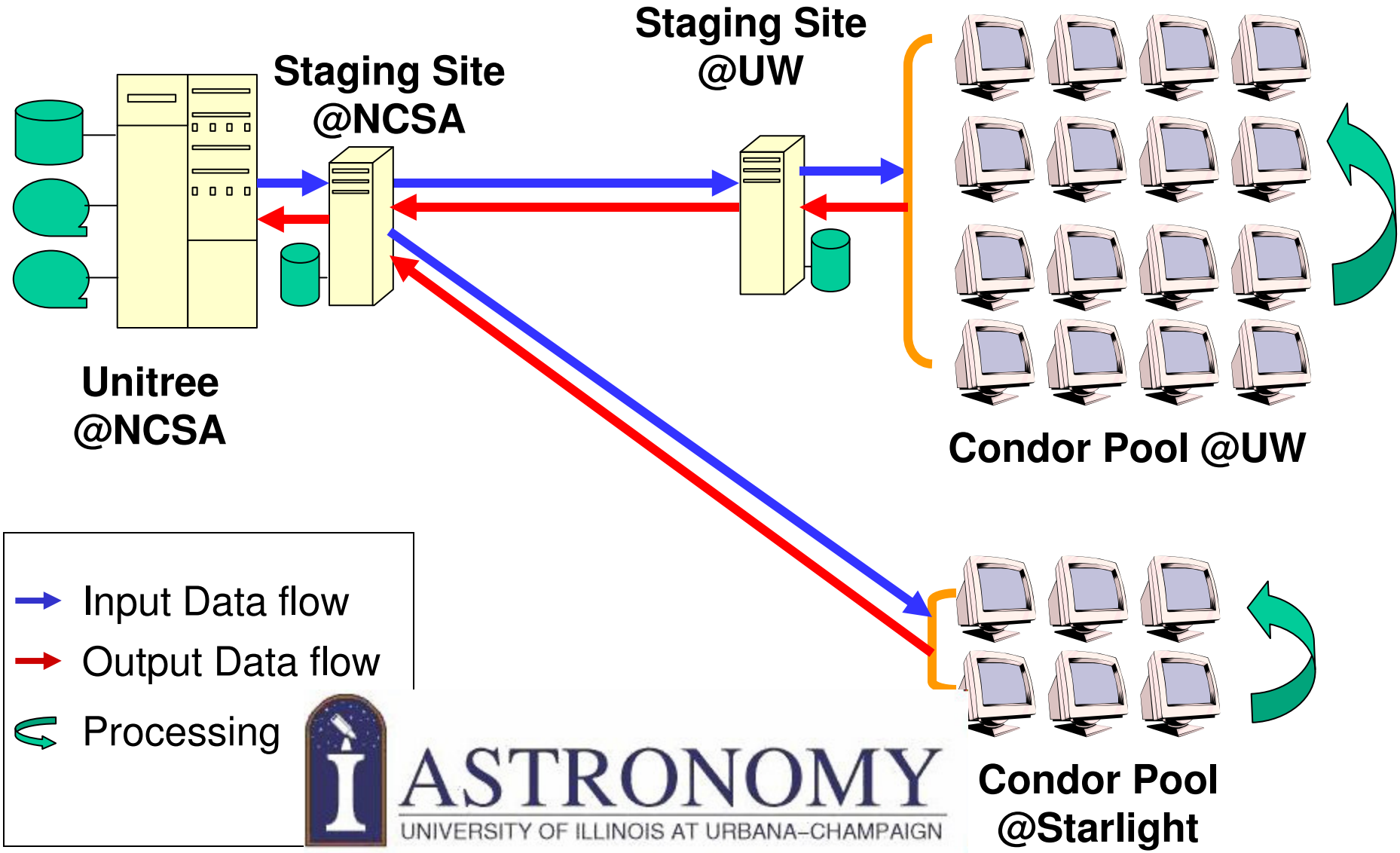
DPOSS Data

NCSA

ASTRONOMY
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

- Palomar-Oschin photographic plates used to map one half of celestial sphere
- Each photographic plate digitized into a single image
- Calibration done by software pipeline at Caltech
- Want to run SExtractor on the images

NCSA Pipeline



NCSA

NCSA Pipeline

- Moved & Processed 3 TB of DPOSS image data in under 6 days
  - Most powerful astronomy data processing facility!
- Adapt for other datasets (Petabytes): Quest2, CARMA, NOAO, NRAO, LSST

And what about

Provenance?

What is Provenance?

prov·e·nance (pròv[1]e-nens, -näns´) *noun*

1. Place of origin; derivation.
2. Proof of authenticity or of past ownership. Used of art works and antiques.

How Can Condor Help?

- User Log – collects "main" events about the life of the job.
- Job record (ClassAd) managed by the submission site.
- DAGMan logs.
- Pre and Post scripts in the DAG.