Pegasus WMS: Leveraging Condor for Workflow Management

Ewa Deelman, Gaurang Mehta, Karan Vahi, Gideon Juve, Mats Rynge, Prasanth Thomas, Jens Voeckler

USC Information Sciences Institute

Miron Livny, Kent Wenger, and others

University of Wisconsin Madison

Funded by the NSF OCI SDCI project

http://pegasus.isi.edu

Examples of Applications

- **Providing a service to a community (Montage project)**
  - Data and derived data products available to a broad range of users
  - A limited number of small computational requests can be handled locally
  - For large numbers of requests or large requests need to rely on shared cyberinfrastructure resources
  - On-the fly analysis generation, portable analysis definition
- **Supporting community-based analysis (SCEC project)**
  - Codes are collaboratively developed
  - Codes are “strung” together to model complex systems
  - Ability to correctly connect components, scalability
- **Processing large amounts of shared data on shared resources (LIGO project)**
  - Data captured by various instruments and cataloged in community data registries.
  - Amounts of data necessitate reaching out beyond local clusters
  - Automation, scalability and reliability
- **Automating the work of one scientist (SIPHT Project, Broad Institute, Epigenomic project, USC)**
  - Data collected in a lab needs to be analyzed in several steps
  - Automation, efficiency, and flexibility (scripts age and are difficult to change)
  - Need to have a record of how data was produced

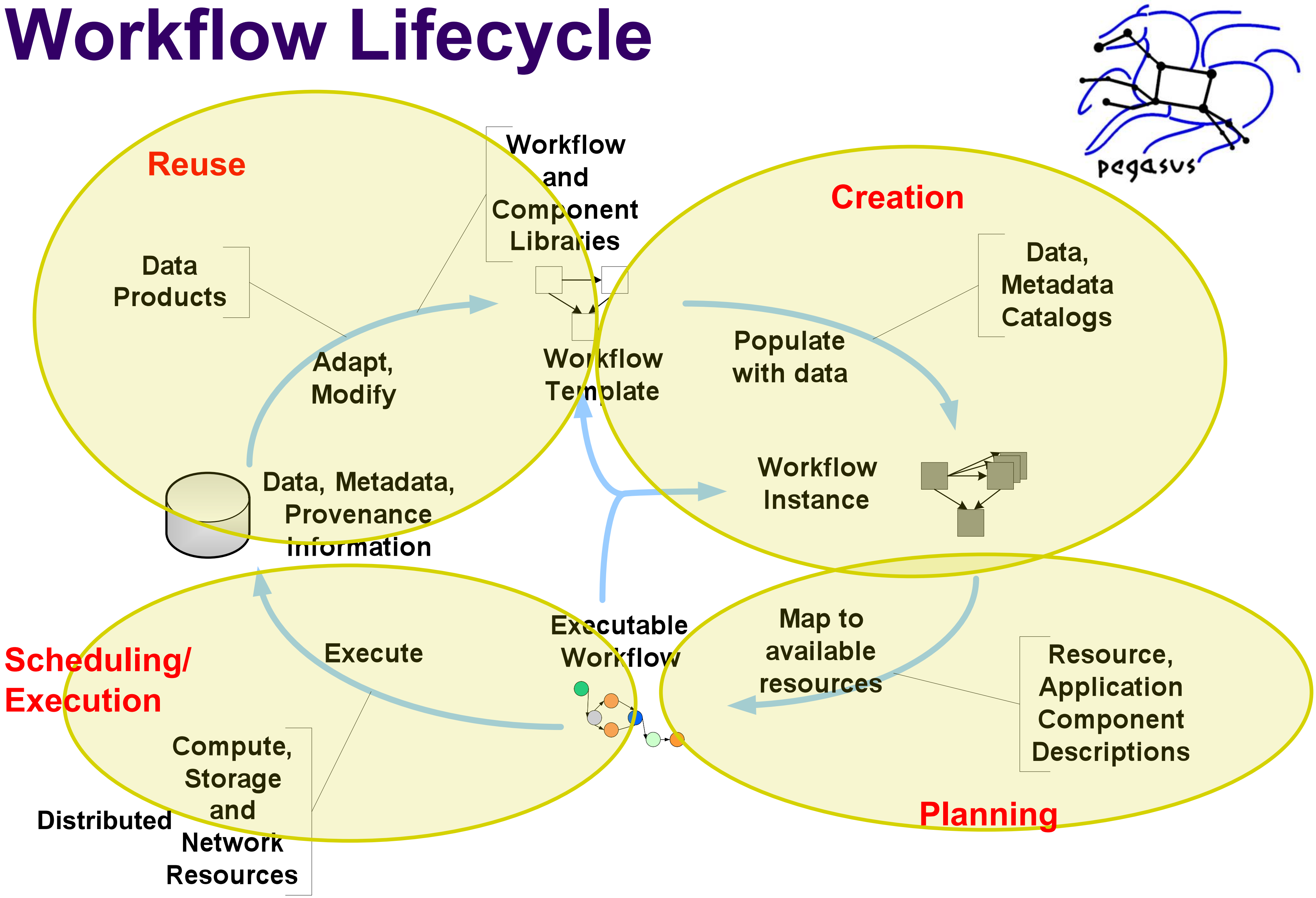
Reasons to use scripts to represent analysis

- You can script something in an afternoon
- You can submit a job directly to a pbs queue or Condor pool
- You can look at stderr to see what went wrong
- You can add calls to measure performance
- You don’t need to learn another language or system

Why Scientific Workflows?

- Workflows can be portable across platforms and scalable
- Workflows are easy to reuse
- Can be shared with others
  - Gives a leg-up to new staff, GRAs, PostDocs, etc
- Workflow Management Systems (WMS) can help recover from failures and optimize overall application performance
- WMS can capture provenance and performance information
- WMS can leverage debugging and monitoring tools

Workflow Lifecycle



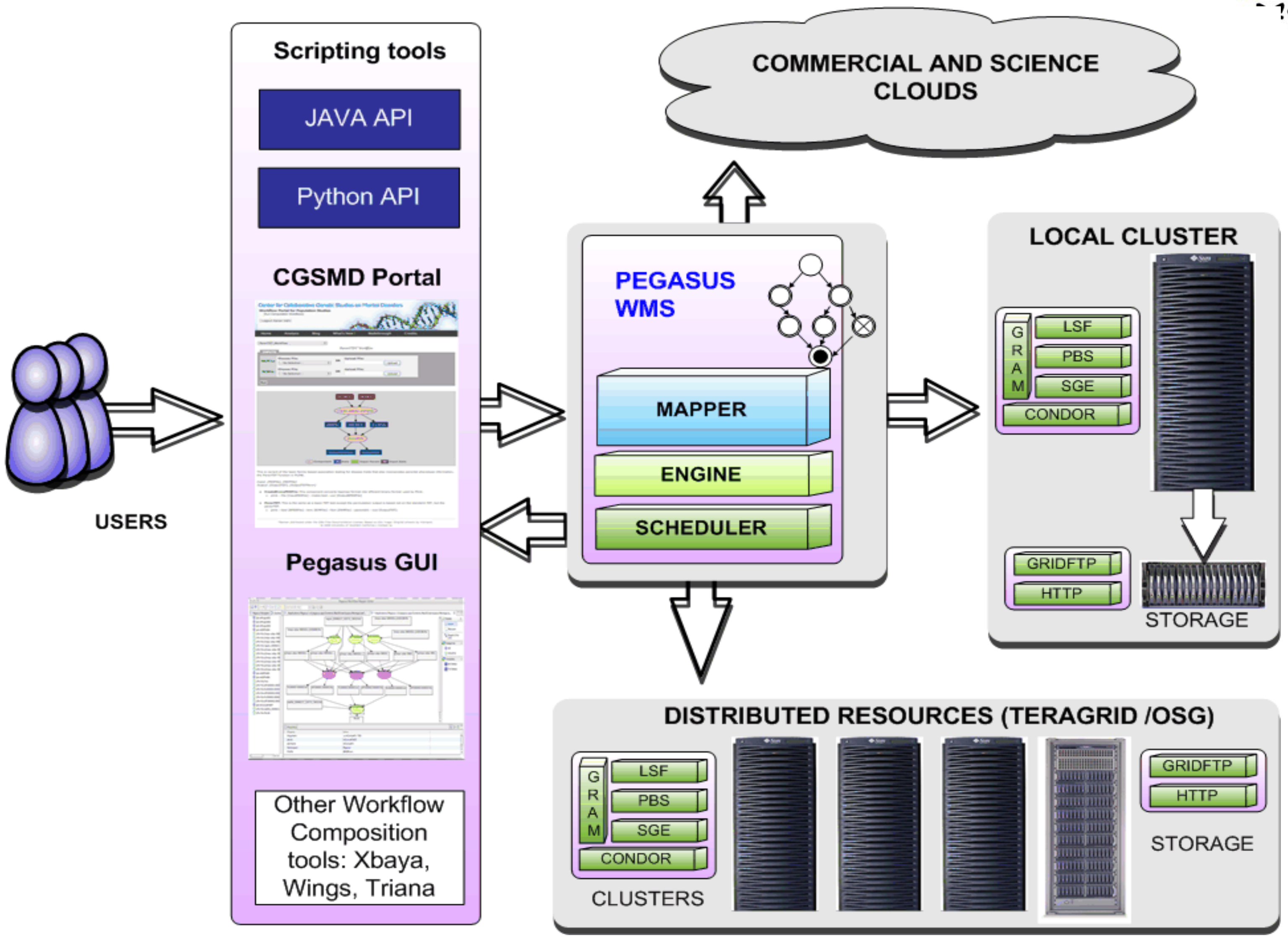
Our Philosophy

- Work closely
  - with users to improve software, make it relevant
  - with CS colleagues to develop new capabilities, share ideas, and develop complex systems
- Users
- Enable them to author workflows in a way comfortable for them
- Allow users to enter the system at any point
- Provide reliability, scalability, performance
- Software
- Be a “good” CI ecosystem member
  - Focus on one aspect of the problem and contribute solutions
  - Leverage existing solutions where possible
- Execution Environment
- Use whatever we can, support heterogeneity

Our Approach

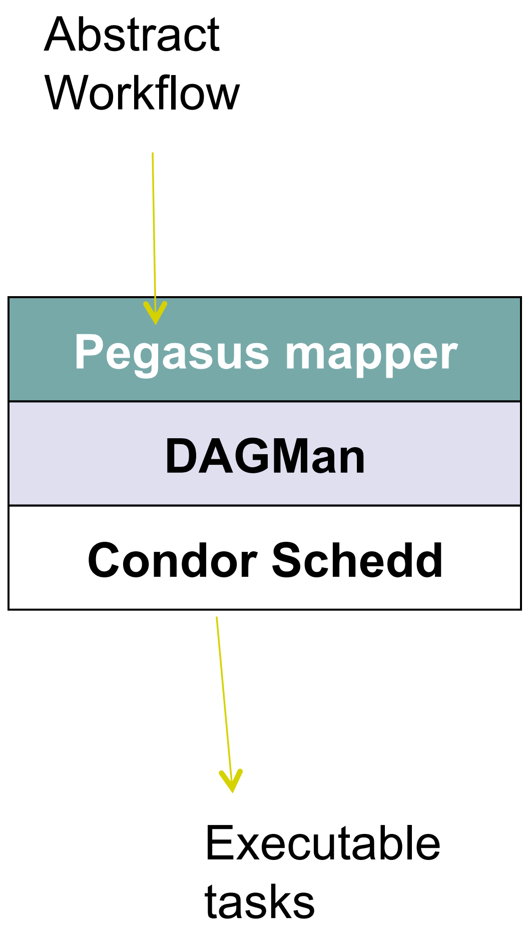
- Representation
  - Support a declarative representation for the workflow (dataflow)
  - Represent the workflow structure as a Directed Acyclic Graph (DAG)
  - Use recursion to achieve scalability
- System
  - Layered architecture, each layer is responsible for a particular function
  - Mask errors at different levels of the system
  - Modular, composed of well-defined components, where different components can be swapped in
  - Open—provides a number of interfaces to enter the system, and exposes interfaces to other CI entities
  - Use and adapt existing graph and other relevant algorithms

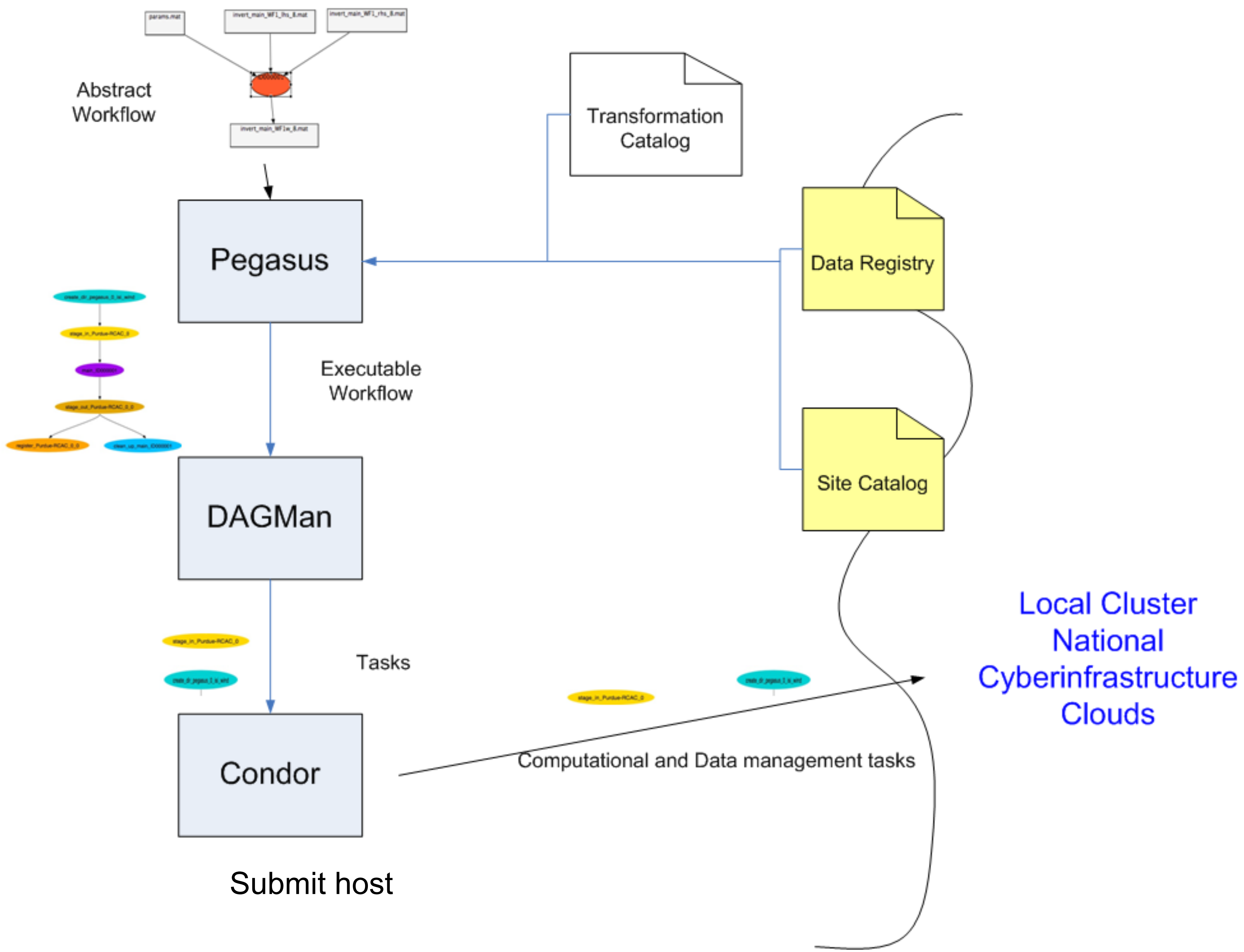
Our system, Pegasus WMS



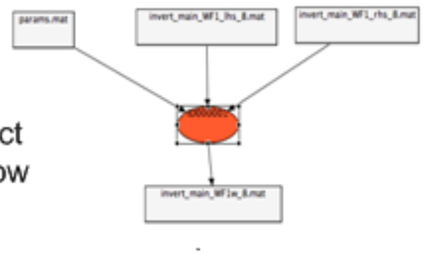
Pegasus WMS, layering functionality

- Condor Schedd
  - A robust task management and execution capability
- DAGMan
  - A workflow executor
  - Scalable and reliable execution of an executable workflow, adaptivity
- Pegasus Mapper
  - a workflow “compiler”
  - target language - DAGMan’s DAG and Condor submit files
    - Generated an executable workflow
  - transforms the workflow for performance and reliability
- Abstract Workflows
  - identifies only the computations that a user wants to do
  - devoid of resource descriptions
  - devoid of data locations

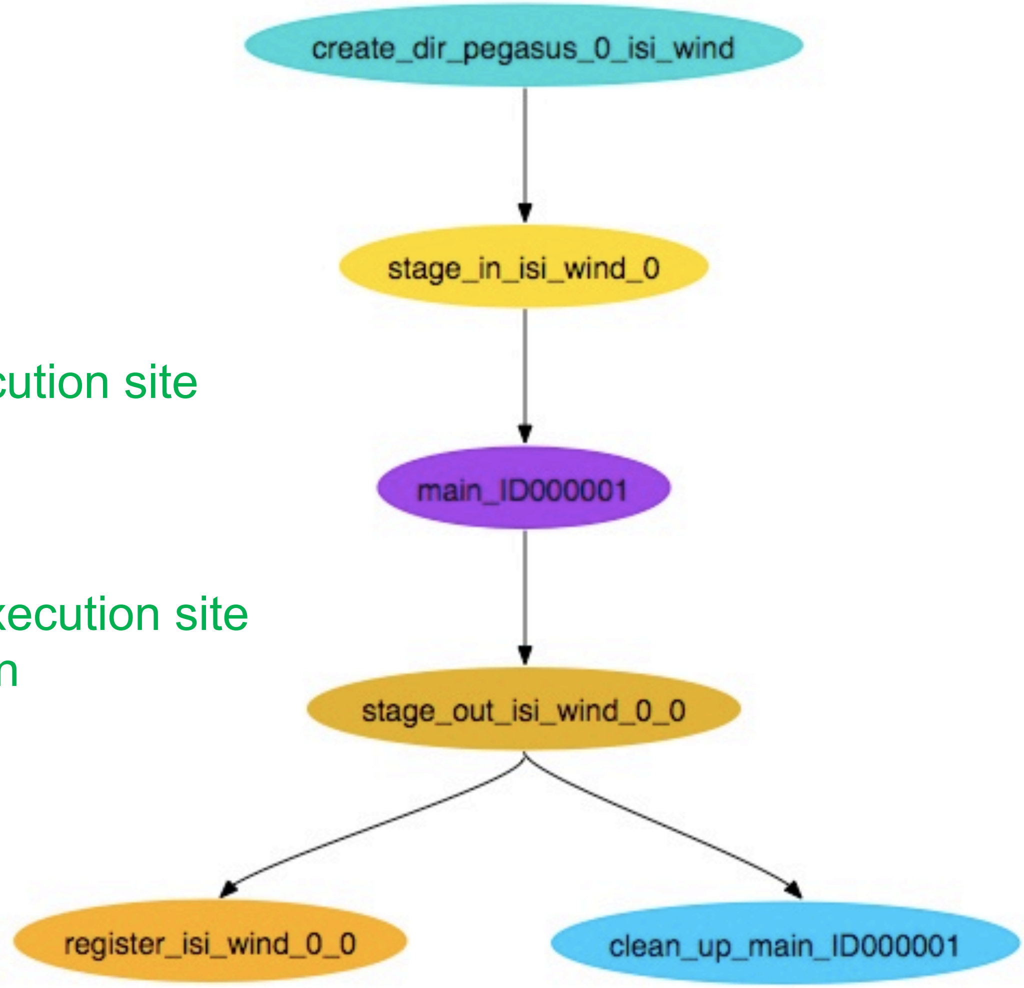



Abstract Workflow
Transformation Catalog
Pegasus
Data Registry
Executable Workflow
DAGMan
Site Catalog
Local Cluster
National
Cyberinfrastructure
Clouds
Tasks
Condor
Computational and Data management tasks
Submit host

Abstract Workflow



Executable Workflow Generated by Pegasus



Pegasus:

Selects an execution site

Selects a data archive

Creates a workflow that

- Creates a "sandbox" on the execution site
- Stages data
- Invokes the computation
- Stages out data
- Registers data and Cleans up execution site
- Captures provenance information

Performs other optimizations

Populated by user or community

Transformation Catalog

```
#
# Transformation Catalog for MPS codes
#

#
# static binaries at submit host
#
local  MPS::main:1.0       gsiftp://submit.isi.edu/mps/main       STATIC_BINARY       INTEL32::LINUX NULL
local  MPS::inversion:1.0  gsiftp://submit.isi.edu/mps/inversion     STATIC_BINARY       INTEL32::LINUX NULL
local   MPS::combine:1.0   gsiftp://submiot.isi.edu/mps/combine     STATIC_BINARY       INTEL32::LINUX NULL

#
# binaries already installed
#
Purdue-RCAC  MPS::main:1.0   /nfs/home/inversion/COMPILE/main INSTALLED INTEL64::LINUX  NULL
Purdue-RCAC  MPS::inversion:1.0   /nfs/home/inversion/COMPILE/inversion INSTALLED INTEL64::LINUX  pegasus::bundle=10
Purdue-RCAC  MPS::combine:1.0   /nfs/home/inversion/COMPILE/combine INSTALLED INTEL64::LINUX  NULL
```

Populated automatically through pegasus-get-sites* or by the user

```xml
<site  handle="Purdue-RCAC" arch="x86" os="LINUX">
    <grid  type="gt2" contact="osg.rcac.purdue.edu:2119/jobmanager-fork" scheduler="Fork" jobtype="auxillary"/>
    <grid  type="gt2" contact="osg.rcac.purdue.edu:2119/jobmanager-condor" scheduler="Condor" jobtype="compute"/>
    <head-fs>
        <scratch>
            <shared>
                <file-server protocol="gsiftp" url="gsiftp://osg.rcac.purdue.edu" mount-point="/scratch/osg">
                </file-server>
                <internal-mount-point mount-point="/scratch/osg"/>
            </shared>
        </scratch>
        <storage>
            <shared>
                <file-server protocol="gsiftp" url="gsiftp://osg.rcac.purdue.edu" mount-point="/scratch/osg">
                </file-server>
                <internal-mount-point mount-point="/scratch/osg"/>
            </shared>
        </storage>
    </head-fs>
    <replica-catalog  type="LRC" url="rlsn://dummyValue.url.edu">
    </replica-catalog>
    <profile namespace="env" key="app" >/apps/osg</profile>
    <profile namespace="env" key="data" >/scratch/osg</profile>
    <profile namespace="env" key="tmp" >/scratch/osg</profile>
    <profile namespace="env" key="wntmp" >/tmp</profile>
</site>
```

DAX snippet

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- generated: 2009-02-27T10:49:29-08:00 -->
<!-- generated by: vahi [??] -->
<adag xmlns="http://pegasus.isi.edu/schema/DAX" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://pegasus.isi.edu/schema/DAX http://pegasus.isi.edu/schema/dax-2.1.xsd"
    version="2.1" count="1" index="0" name="pegasus" jobCount="1" fileCount="0" childCount="0">
<!-- part 1: list of all referenced files (may be empty) -->
<!-- part 2: definition of all jobs (at least one) -->
  <job id="ID000001" namespace="MPS" name="main" version="1.0">
    <argument>-a main -T60-i <filename file="params.mat"/> <filename file="invert_main_WF1_lhs_8.mat"/>
      <filename file="invert_main_WF1_rhs_8.mat"/>
        -o <filename file="invert_main_WF1w_8.mat"/>  </argument>
    <profile namespace="ENV" key="HELIO_HOME">.</profile>
    <uses file="params.mat" link="input" register="true" transfer="true" type="data"/>
    <uses file="invert_main_WF1_lhs_8.mat" link="input" register="true" transfer="true
    <uses file="invert_main_WF1_rhs_8.mat" link="input" register="true" transfer="true
    <uses file="invert_main_WF1w_8.mat" link="output" register="true" transfer="true"
  </job>
<!-- part 3: list of control-flow dependencies (may be empty) -->
</adag>
```

Pegasus

CondorDAG / Condor Submit files

```
########################################################################
# PEGASUS WMS GENERATED DAG FILE
# DAG invert
# Index = 0, Count = 1
########################################################################

JOB inversion_ID0000002 inversion_ID0000001.sub
SCRIPT POST inversion_ID00000081 /lfs1/bin/exitpost  inversion_ID0000001.out
RETRY inversion_ID00000081 3
...

PARENT main_ID0000001 CHILD inversion_ID0000002
PARENT stage_out_Purdue-RCAC_0_0 CHILD register_Purdue-RCAC_0_0
PARENT stage_in_Purdue-RCAC_0 CHILD main_ID0000001

########################################################################
# End of DAG
########################################################################
```

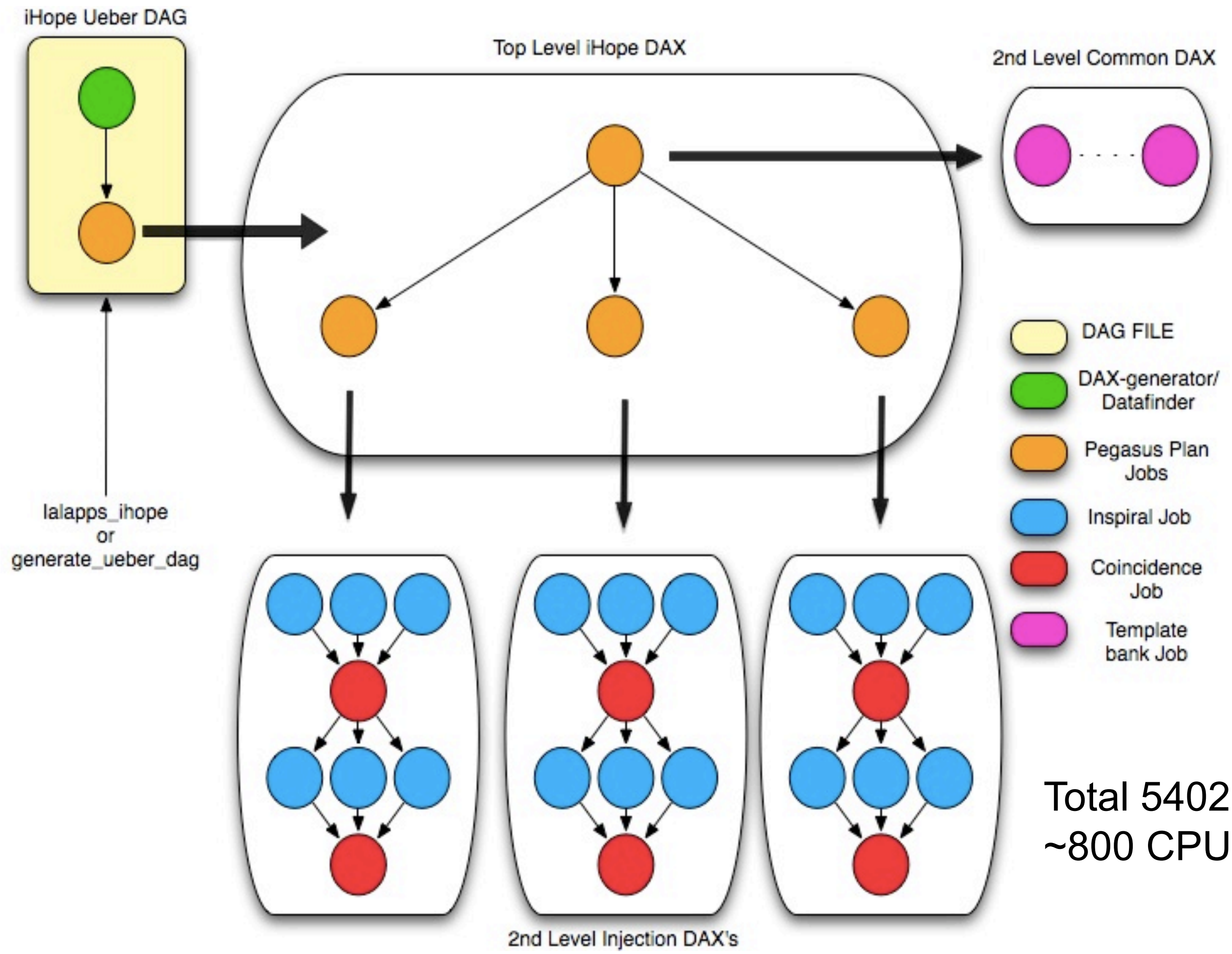
```
######################################################################
# PEGASUS WMS GENERATED SUBMIT FILE
# DAG : invert, Index = 0, Count = 1
# SUBMIT FILE NAME : main_ID0000001.sub
######################################################################
stream_error  = false
stream_output = false
environment = GLOBUS_LOCATION=/nfs/software/globus/default;
arguments = "-n MPS::main:1.0 -N null -R isi_wind /Codes main parameters"
copy_to_spool = false
error = /lfs1/work/helio/dags/main_ID0000001.err
executable = /nfs/software/pegasus/default/bin/kickstart
globusrsl = (jobtype=single)
globusscheduler = purdue.isi.edu/jobmanager-condor
log = /tmp/invert-020787.log
notification = NEVER
output = /lfs1/work/helio/dags/main_ID0000001.out
periodic_release = (NumSystemHolds <= 3)
periodic_remove = (NumSystemHolds > 3)
remote_initialdir = /nfs/shared-scratch/helio/run0019
submit_event_user_notes = pool:isi_purdue
transfer_error = true
transfer_executable = false
transfer_output = true
universe = globus

######################################################################
# END OF SUBMIT FILE
######################################################################
```

*OSG interface provided by Vikas Patel and Sebastian Goasguen

The LIGO example, migrating up the software stack

- LIGO has been using DAGMan for its scientific analysis
- Issue 1: LIGO users log onto to a particular cluster and launch computations there (no load balance)
- Issue 2: Sometimes part of input data is “vetoed” and needs to be eliminated from the analysis, so potentially large amounts of redundant work need to be redone
- Issue 3: Some tasks are very short running and incur large overheads
- Issue 4: Want to be able to run the same workflow on other Grids (OSG), and share analyses with EU colleagues
- Issue 5: Want to be able to keep parts of a pipeline as a DAG—for legacy visualization pipelines
- Issue 6: For large workflows, it is difficult to analyze the DAGMan/ Condor logs to pinpoint problems

LIGO on OSG and LDG



Total 5402 jobs
~800 CPU hours cumulative

LIGO Issues

- Issue 1: LIGO users log onto to a particular cluster and launch computations there (no load balance)
  - Pegasus uses information services or user-provided information to schedule an entire workflow onto a single cluster or across clusters
  - Pegasus brings back intermediate and final results to a user-specified location
- Issue 2: Sometimes part of input data is "vetoed" and needs to be eliminated from analysis, so potentially large amounts of redundant work need to be redone
  - Pegasus has the concept of "virtual data" where if data are already available it will be reused
  - If the same workflow is re-submitted, and some intermediate data are already available, the executable workflow will reuse it ➔ efficient execution, scientists can start analysis without waiting for final "vetoes"
- Issue 3: Some tasks are very short running and incur large overheads
  - Pegasus can automatically cluster tasks together so that they are treated as one by DAGMan, Condor, and the target execution system

- Issue 4: Want to be able to run on other Grids, and share analyses with EU colleagues
  - Pegasus DAXes are devoid of resource information, so to run a DAX in a new environment, only “local” info about resources and data locations needs to be given separately, Pegasus will generate the right DAG and Condor Submit files

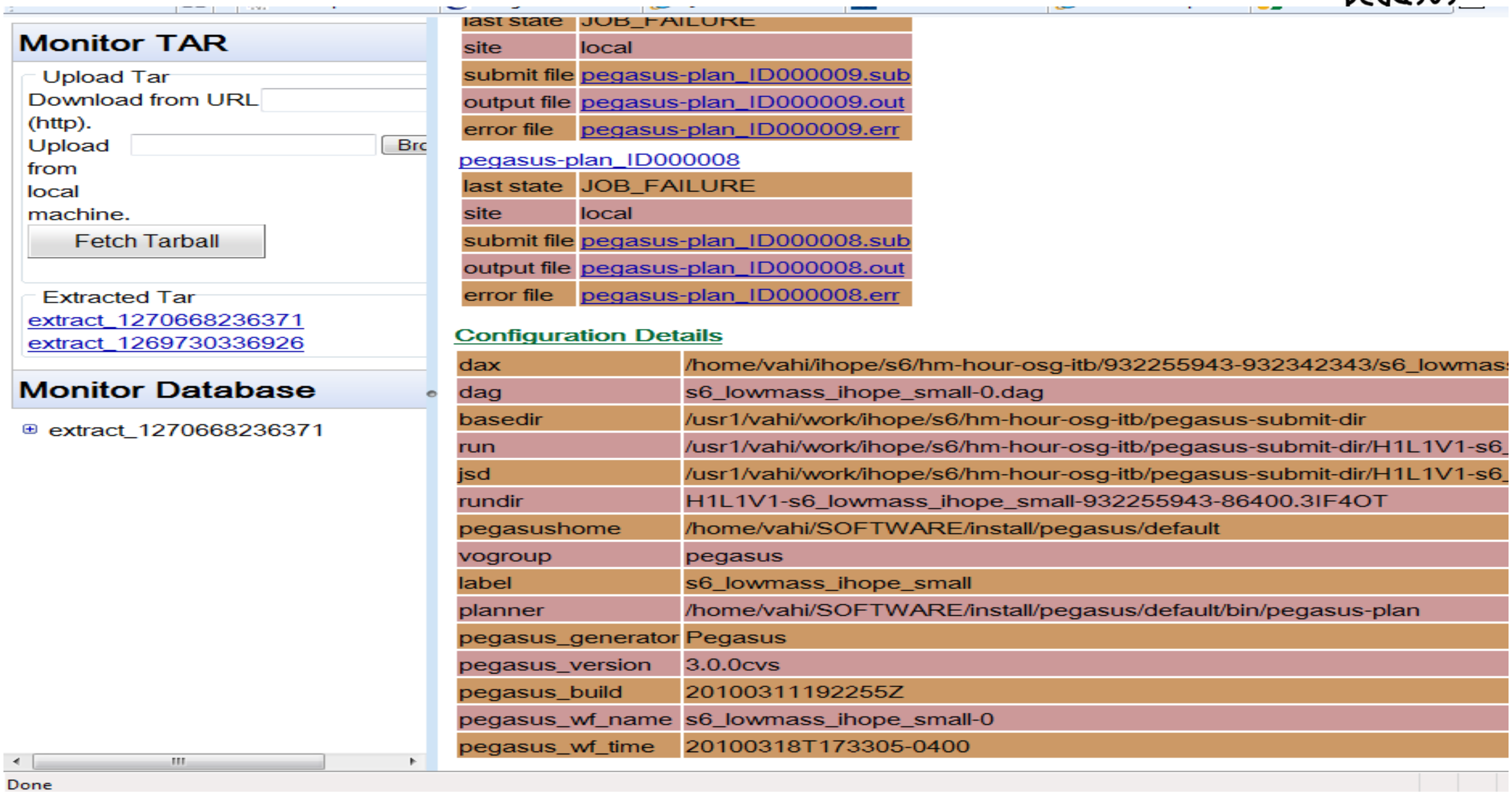
- Issue 5: Want to be able to keep parts of a pipeline as a DAG—legacy visualization pipelines
  - You can embed a DAG into a DAX and this information will be passed through to DAGMan ➔ You can use any DAGMAN features inside a DAX

- Issue 6: Difficulty analyzing the DAGMan/Condor logs to pinpoint problems
  - Developed pegasus-analyzer that can traverse
    - the DAGMan.out and Condor's *.err and *.out information

"This is so much easier!" -- Duncan Brown, LIGO

```
===================lalapps_tmpltbank_ID002291============================
  last state: JOB_FAILURE
       site: local
submit file: /usr1/ilya/log/H1L1V1- s6_highmass_ihope-937800015-4197585.3CpZuA/datafind/
       lalapps_tmpltbank_ID002291.sub
output file: /usr1/ilya/log/H1L1V1-s6_highmass_ihope-937800015-4197585.3CpZuA/datafind/lalapps_tmpltbank_ID002291.out
error file: /usr1/ilya/log/H1L1V1- s6_highmass_ihope-937800015-4197585.3CpZuA/datafind/lalapps_tmpltbank_ID002291.err

------------------------ lalapps_tmpltbank_ID002291.out------------------------
 ------------------------lalapps_tmpltbank_ID002291.err------------------------

XLAL Error - XLALFrNext: gap in frame data
XLAL Error - XLALFrNext: time 941096000.000000 is end of frame 3999 of file URL
       file://localhost/frames/VSR2/HrecOnline/V1/V-HrecOnline-941/V-HrecOnline-941092000-4000.gwf
XLAL Error - XLALFrNext: time 941100000.000000 is start of frame 0 of  file URL
       file://localhost/frames/VSR2/HrecOnline/V1/V-HrecOnline-941/V-HrecOnline-941100000-4000.gwf
XLAL Error - XLALFrNext (FrameStream.c:608): Invalid time
Error[2] 8192: function LALFrNext, file FrameStream.c, line 1046, $Id$
ABORT: Gap in the data

……………………..
```

- Developing a browser-based visualization for performance and failure analysis



- "When LIGO inspiral group switched the from DAGs to DAXes—we did not notice, the results were delivered as before" -- Frederique Marion, LIGO-Virgo CBC Group

Challenges in workflow reliability
leveraging the software layers

- Resources fail
  - Provide a retry mechanism
- Services fail (data movement, data registration)
  - Retry the action, choose a different service
- Computations fail within a workflow
  - Checkpoint the workflow
- Storage gets filled up
  - Analyze the workflow and clean up unneeded data as the workflow execution progresses

NMI Test and Build Lab

Production releases

Nightly builds and tests

- 3 Pegasus packages (Mapper, WMS, Worker)
- 15 platforms
- .tar.gz / .deb / .rpm
  - Latest code is pulled from the Pegasus SVN, built and tested.
  - Generated packages (~50) are automatically pushed back to the Pegasus website

Pinned Condor release build used as input to the WMS package

The latest stable release of PEGASUS WMS is 2.4.1

Read the RELEASE NOTES.

The releases work with JDK or JRE 1.5+

The following packages are available

- Packages with Condor
- Packages without Condor
- Worker Packages
- Source Packages (without Condor)

Pegasus With Condor

DEBIAN	5.0(x86_64)	5.0(x86)	4.0(x86)
RHEL/CentOS/SL	5.x(x86_64)	5.x(x86)	
OSX	Tiger(x86)		

Pegasus Without Condor (if you already have condor)

DEBIAN	5.0(x86_64)	5.0(x86)	4.0(x86_64)	4.0(x86)	
RHEL/CentOS/SL	5.x(x86_64)	5.x(x86)	4.x(x86_64)	4.x(x86)	4.x(ia64)
SUSE	9.0(x86_64)	9.0(x86)	9.0(ia64)		
OSX	Leopard(x86_64)	Tiger(x86)			

Pegasus Worker Packages

DEBIAN	5.0(x86_64)	5.0(x86)	4.0(x86_64)	4.0(x86)	
RHEL/CentOS/SL	5.x(x86_64)	5.x(x86)	4.x(x86_64)	4.x(x86)	4.x(ia64)
SUSE	9.0(x86_64)	9.0(x86)	9.0(ia64)		
OSX	Leopard(x86_64)	Tiger(x86)			

Pegasus Source Packages

Source Tarball

To Check out from SVN

Future Directions

- Debugging workflows is still difficult
  - Need to be able to interpret errors
  - Analyze what happened
  - Need to be able to provide error information at the level needed by the user

- Online monitoring is still an issue for large workflows (teaming up with Netlogger)

- Automatically exploiting data parallelism, how to subdivide a data set

- Generate computational bundles (data, codes, configurations) – automated boinc

Want to try?

pegasus@isi.edu

- Hands-on help

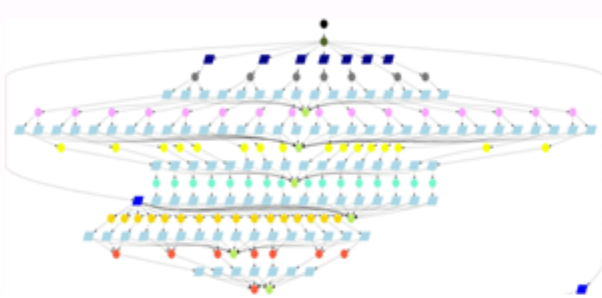
http://pegasus.isi.edu

- Tutorial materials

Page ▾ Saf

Biology	**Earth Sciences**	**Physical Sciences**	**Education**
Bioinformatics	Climate Modeling	Astronomy	Online Classroom
Neuroscience	Earthquake Science	Chemistry	
Botany	Ocean Science	Energy	
Genome Analysis	Limnology	Physics	
		Computer Science	
		Helioseismology	
		Data Mining	

Bioinformatics



DNA sequencing

The USC Epigenome Center is currently using the Illumina Genetic Analyzer (GA) system to generate high throughput DNA sequence data (up to 8 billion nucleotides per week) to map the epigenetic state of human cells on a genome-wide scale.

Epigenomic Workflow (computational jobs are shown as circles, data transfer jobs as rhomboids).

Related Technologies: Corral-WMS Th. pm by Mats Rynge