Science Automation with the Pegasus Workflow Management System

Ewa Deelman

USC Information Sciences Institute

http://www.isi.edu/~deelman

Funding from DOE, NSF, and NIH

The Problem

- Scientific data is being collected at an ever increasing rate
  - The “old days” -- big, focused experiments– LHC
  - Today also “cheap” DNA sequencers – and an increasing number of them
- The complexity of the computational problems is ever increasing
- Local compute resources are often not enough
  - Too small, limited availability
  - Data sets are distributed
- The computing infrastructure keeps changing
  - Hardware, software, but also computational models

Our approach

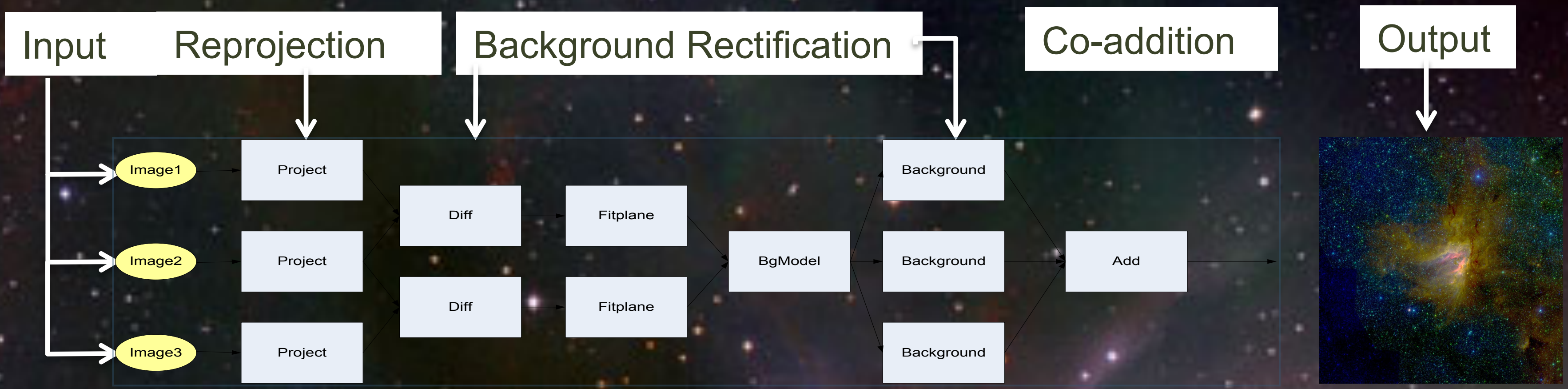
- Provide a way to structure applications in such a way that enables them to be automatically managed
  - In a portable way: same description that works on different resources
  - In a way that scientists can interpret the results
- Develop a system that
  - Maps the application description onto the available resources
  - Manages its execution on heterogeneous resources
  - Sends results back to the user or archive
  - Provides good performance, reliability, scalability

Outline

- Scientific Workflows and Application Examples
- Managing scientific workflows
- Pegasus and its features
- Conclusions

Science-grade Mosaic of the Sky

Science-grade Mosaic of the Sky

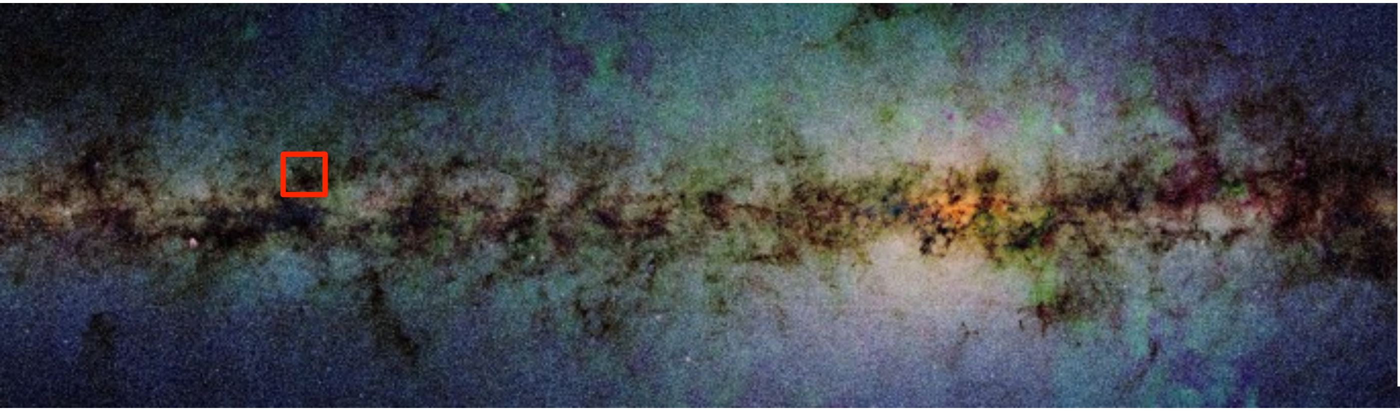


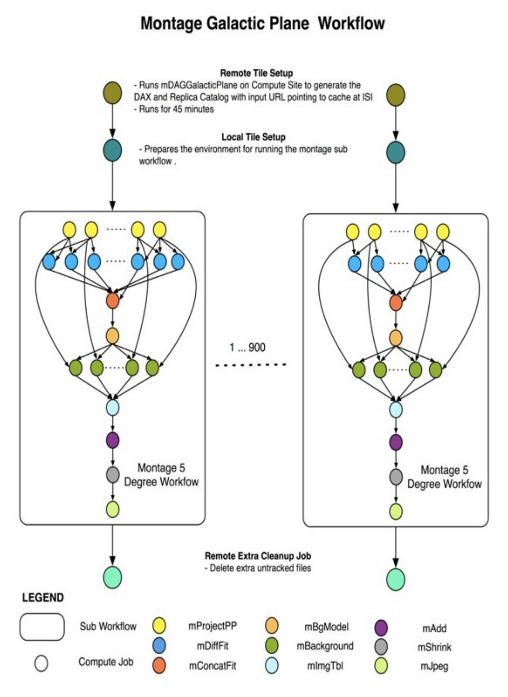
Montage Workflow

Amazon M1 large with 2 cores

Size of mosaic in degrees square	Number of input data files	Number of tasks	Number of intermediate files	Total data footprint	Cummulative wall time
1	84	387	770	1.8 GB	11 mins
2	300	1442	2880	6.4 GB	43 mins
4	685	3738	7466	17 GB	1 hour, 56 mins
6	1461	7462	14904	35 GB	3 hours, 42 mins
8	2565	12757	25480	59 GB	6 hours, 45 mins

Some workflows are large-scale and data-intensive




Montage Galactic Plane Workflow
Remote Tile Setup
- Runs mDAGGalacticPlane on Compute Site to generate the DAX and Replica Catalog with input URL pointing to cache at ISI
- Runs for 45 minutes
Local Tile Setup
- Prepares the environment for running the montage sub workflow .
1 ... 900
Montage 5 Degree Workfow
Montage 5 Degree Workfow
Remote Extra Cleanup Job
- Delete extra untracked files
LEGEND
Sub Workflow
Compute Job
mProjectPP
mDiffFit
mConcatFit
mBgModel
mBackground
mImgTbl
mAdd
mShrink
mJpeg

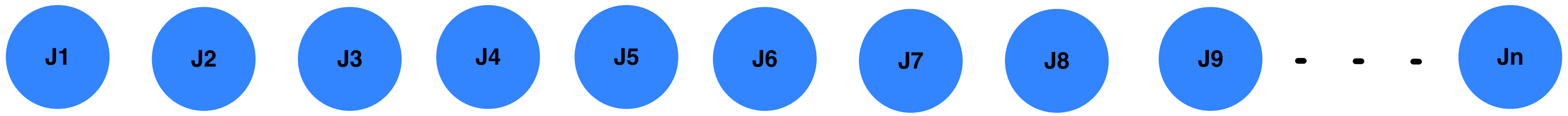
John Good (Caltech)

- **Montage Galactic Plane Workflow**
  - **18 million input images (~2.5 TB)**
  - **900 output images (2.5 GB each, 2.4 TB total)**
  - **10.5 million tasks (34,000 CPU hours)**

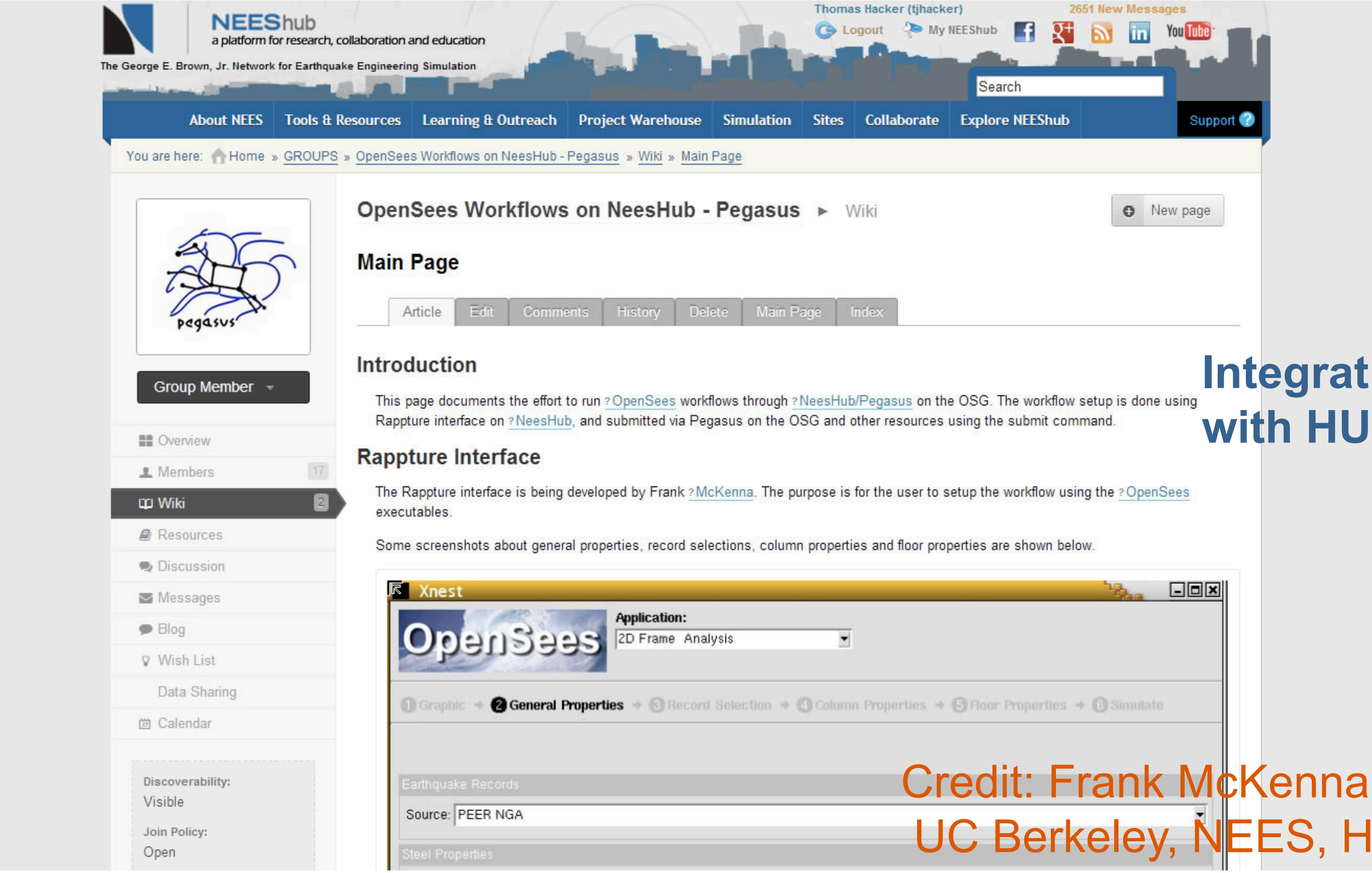
× 17

- **Need to support hierarchical workflows and scale**

Workflows can be simple!



Sometimes you want to “hide” the workflow



NEEShub
a platform for research, collaboration and education
The George E. Brown, Jr. Network for Earthquake Engineering Simulation

Thomas Hacker (tjhacker) 2651 New Messages
Logout My NEEShub

Search

You are here: Home » GROUPS » OpenSees Workflows on NeesHub - Pegasus » Wiki » Main Page

OpenSees Workflows on NeesHub - Pegasus ▸ Wiki

Discoverability:
Visible
Join Policy:
Open

Main Page

Introduction

This page documents the effort to run ?OpenSees workflows through ?NeesHub/Pegasus on the OSG. The workflow setup is done using Rappture interface on ?NeesHub, and submitted via Pegasus on the OSG and other resources using the submit command.

Rappture Interface

The Rappture interface is being developed by Frank ?McKenna. The purpose is for the user to setup the workflow using the ?OpenSees executables.

Some screenshots about general properties, record selections, column properties and floor properties are shown below.

Application: 2D Frame Analysis

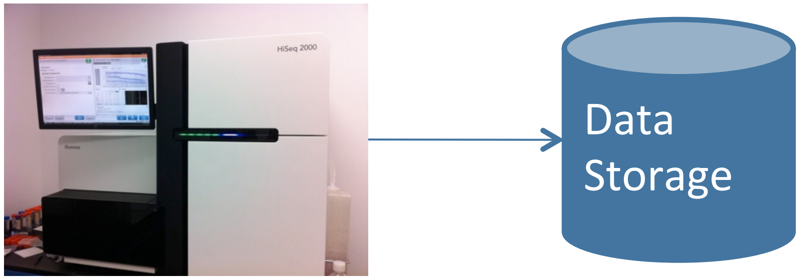
1 Graphic → 2 General Properties → 3 Record Selection → 4 Column Properties → 5 Floor Properties → 6 Simulate

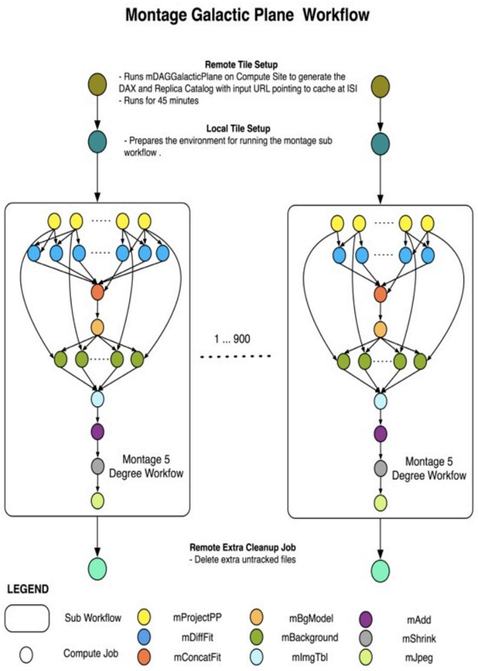
Earthquake Records
Source: PEER NGA
Steel Properties

Integration with HUBzero

Credit: Frank McKenna
UC Berkeley, NEES, HUBzero

Sometimes the environment is complex




Montage Galactic Plane Workflow
Remote Tile Setup
- Runs mDAGGalacticPlane on Compute Site to generate the DAX and Replica Catalog with input URL pointing to cache at ISI
- Runs for 45 minutes
Local Tile Setup
- Prepares the environment for running the montage sub workflow .
1 ... 900
Montage 5 Degree Workflow
Montage 5 Degree Workflow
Remote Extra Cleanup Job
- Delete extra untracked files
LEGEND
Sub Workflow
Compute Job
mProjectPP
mDiffFit
mConcatFit
mBgModel
mBackground
mImgTbl
mAdd
mShrink
mJpeg

Work definition

Local Resource

Campus Cluster

XSEDE

NERSC

ALCF

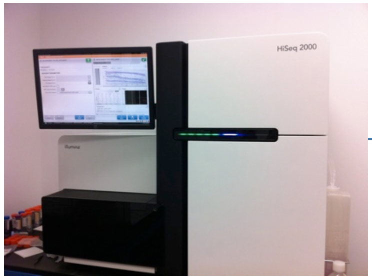
OLCF

Open Science Grid

FutureGrid

Amazon Cloud

Sometime you want to change or combine resources



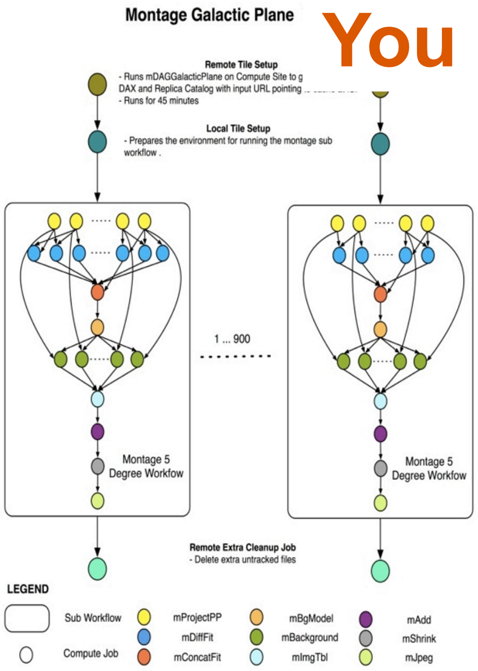
Data Storage

data

Campus Cluster

XSEDE

You don’t want to recode your workflow



ALCF

OLCF

Open Science Grid

work

FutureGrid

Amazon Cloud

Local Resource

Workflow Management

- Assume a high-level workflow specification
- Assume the potential use of different resources within a workflow or over time
  - Need a planning capability to map from high-level to executable workflow
  - Need to manage the task dependencies
  - Need to manage the execution of tasks on the remote resources
- Need to provide provenance information
- Need to provide scalability, performance, reliability

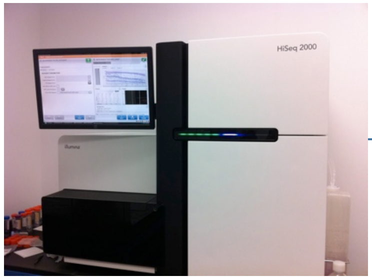
Outline

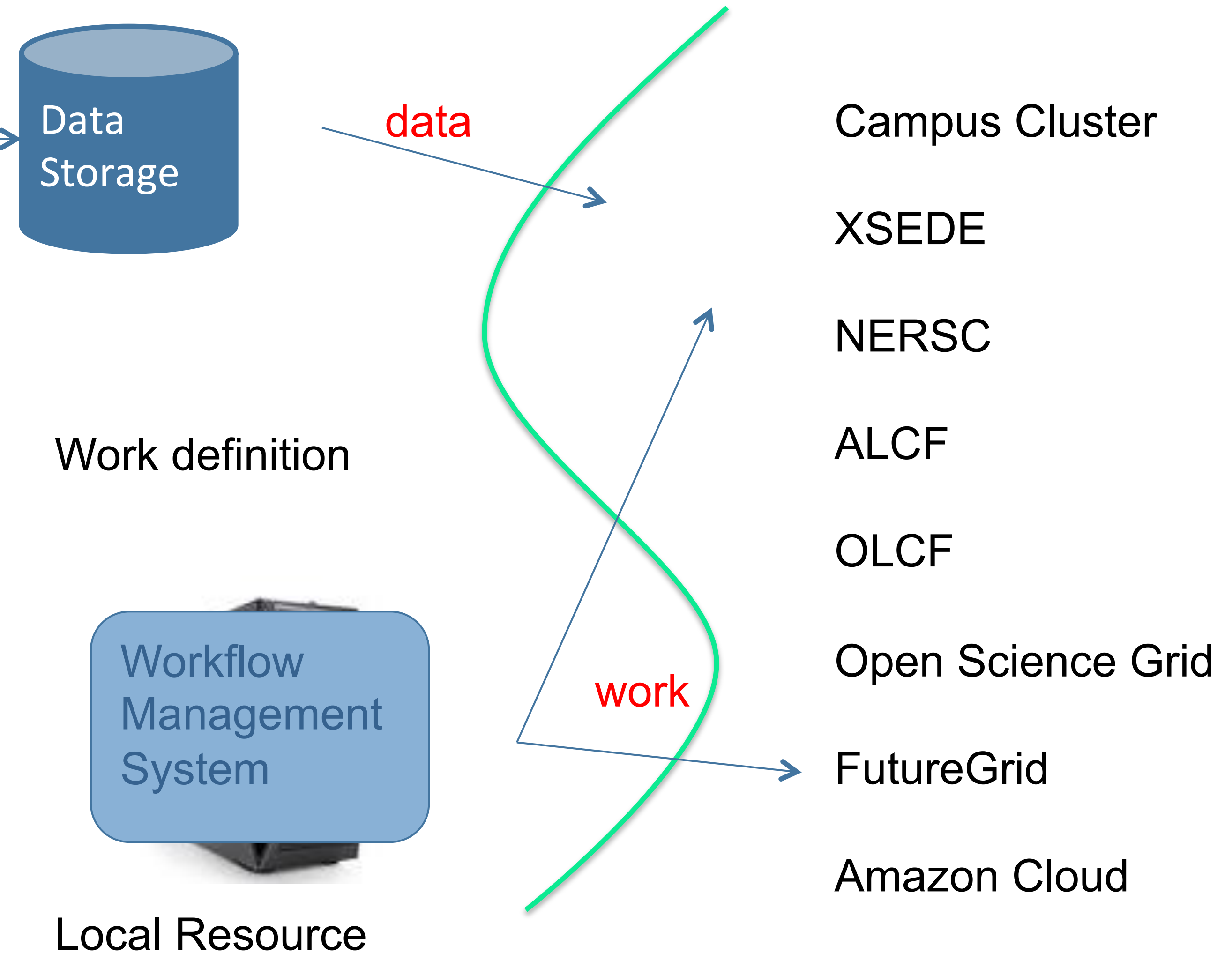
- Scientific Workflows and Application Examples
- Managing scientific workflows
- Pegasus and its features
- Conclusions

Our Approach

- **Analysis Representation**
  - Support a declarative representation for the workflow
  - Represent the workflow structure as a Directed Acyclic Graph (DAG)
  - Tasks operate on files
  - Use recursion to achieve scalability
- **System (Plan for the resources, Execute the Plan, Manage tasks)**
  - 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
  - Use and adapt existing graph and other relevant algorithms

Submit locally, compute Globally





Work definition

Local Resource

Campus Cluster

XSEDE

NERSC

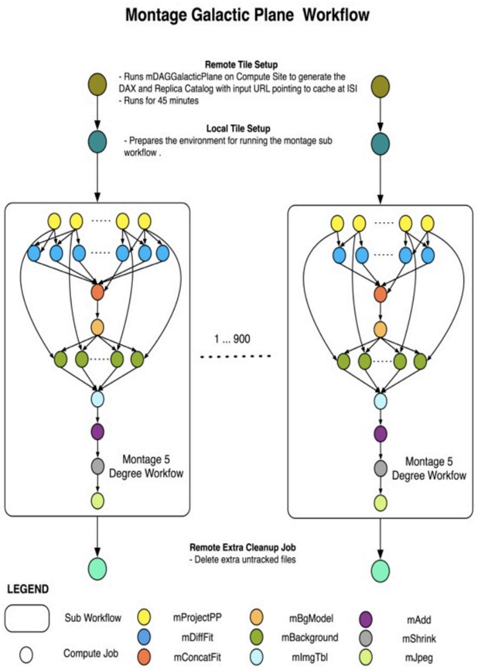
ALCF

OLCF

Open Science Grid

FutureGrid

Amazon Cloud



- Runs mDAGGalacticPlane on Compute Site to generate the DAX and Replica Catalog with input URL pointing to cache at ISI
- Runs for 45 minutes

- Prepares the environment for running the montage sub workflow .

- Delete extra untracked files

Pegasus
Workflow Management System (est. 2001)

- A collaboration between USC and the Condor Team at UW Madison
- Maps a resource-independent “abstract” workflow onto resources and executes the “concrete” workflow
- Used by a number of applications in a variety of domains
- Provides reliability—can retry computations from the point of failure
- Provides scalability—can handle large data and many computations (kbytes-TB of data, 1-10^6 tasks)
- Infers data transfers, restructures workflows for performance
- Automatically captures provenance information
- Can run on resources distributed among institutions, laptop, campus cluster, Grid (OSG, XSEDE), Cloud (Amazon, FutureGrid)

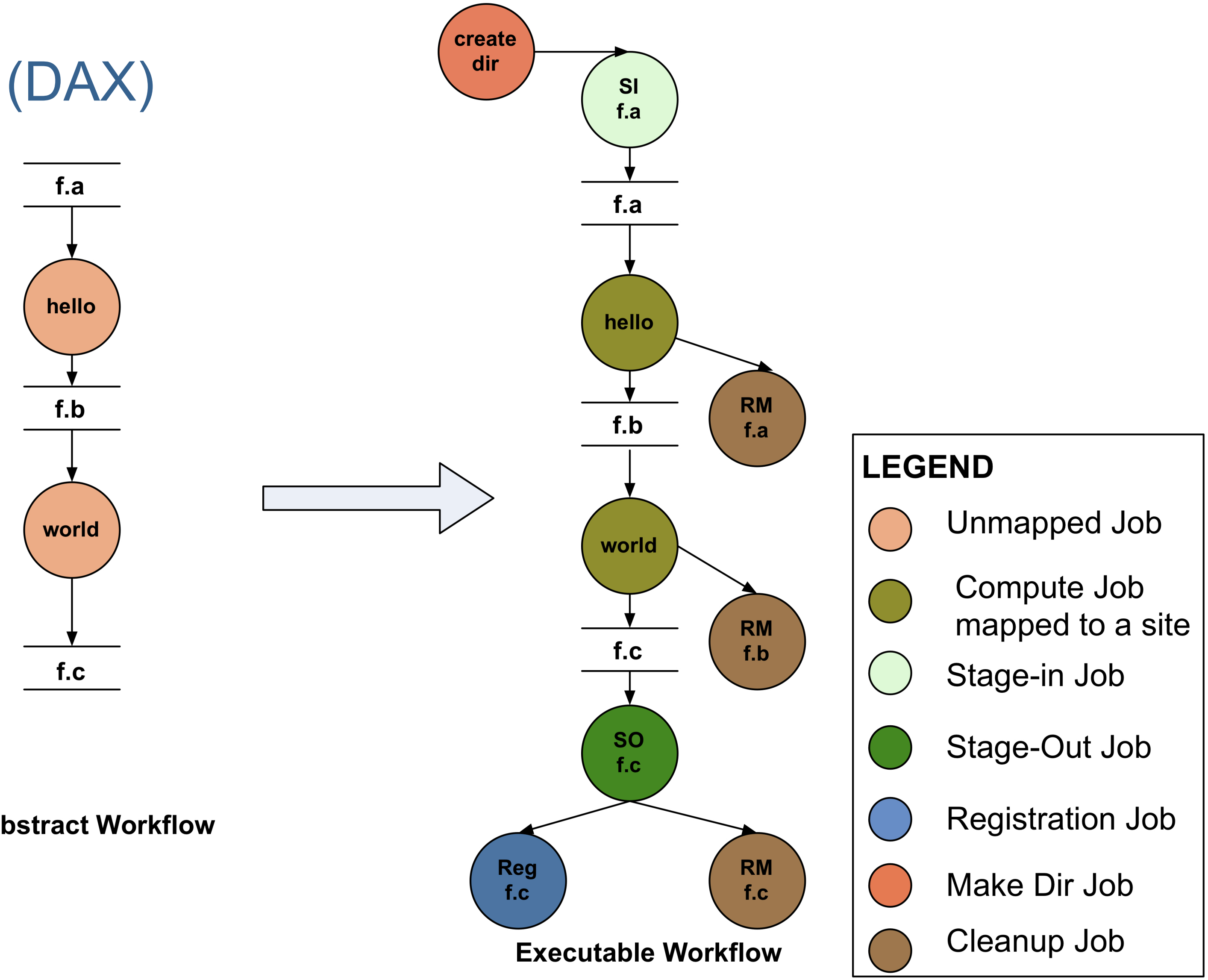
Pegasus Workflow Management System

- A workflow "compiler"
  - Input: abstract workflow description, resource-independent
  - Auxiliary Info (catalogs): available resources, data, codes
  - Output: executable workflow with concrete resources
  - Automatically locates physical locations for both workflow tasks and data
  - Transforms the workflow for performance and reliability
- A workflow engine (DAGMan)
  - Executes the workflow on local or distributed resources (HPC, clouds)
  - Task executables are wrapped with *pegasus-kickstart* and managed by Condor *schedd*
- Monitoring tools
- Provenance and execution traces collection

Generating executable workflows

APIs for workflow specification (DAX--- DAG in XML)

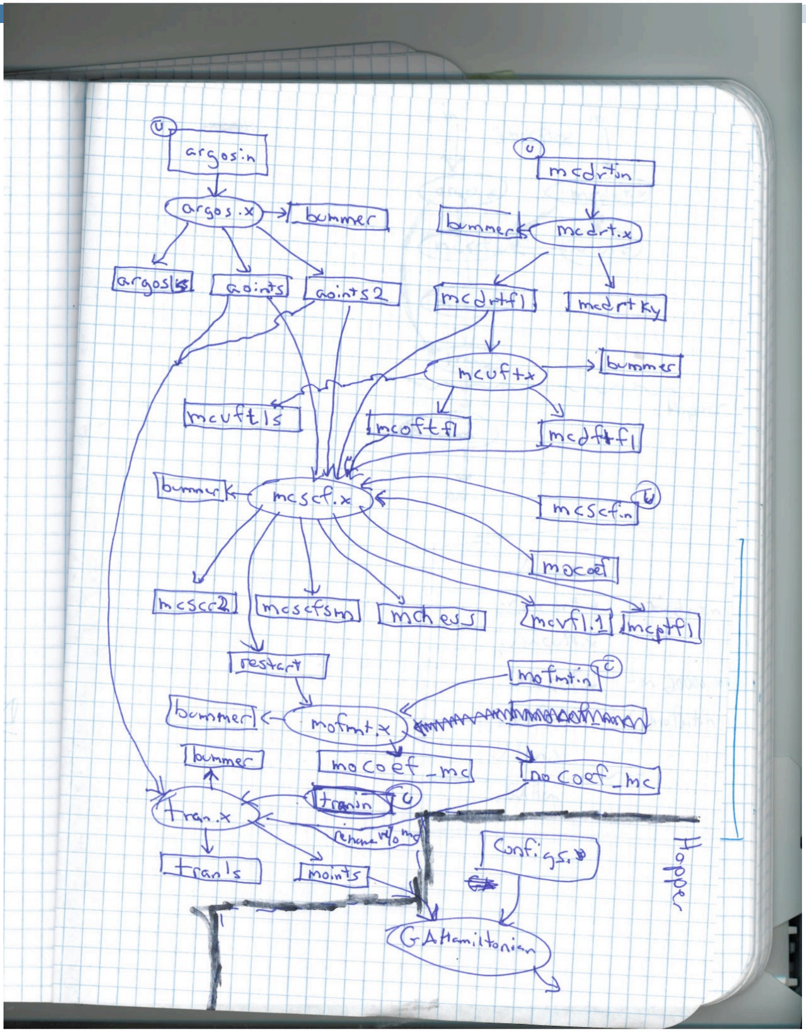
Java, Perl, Python

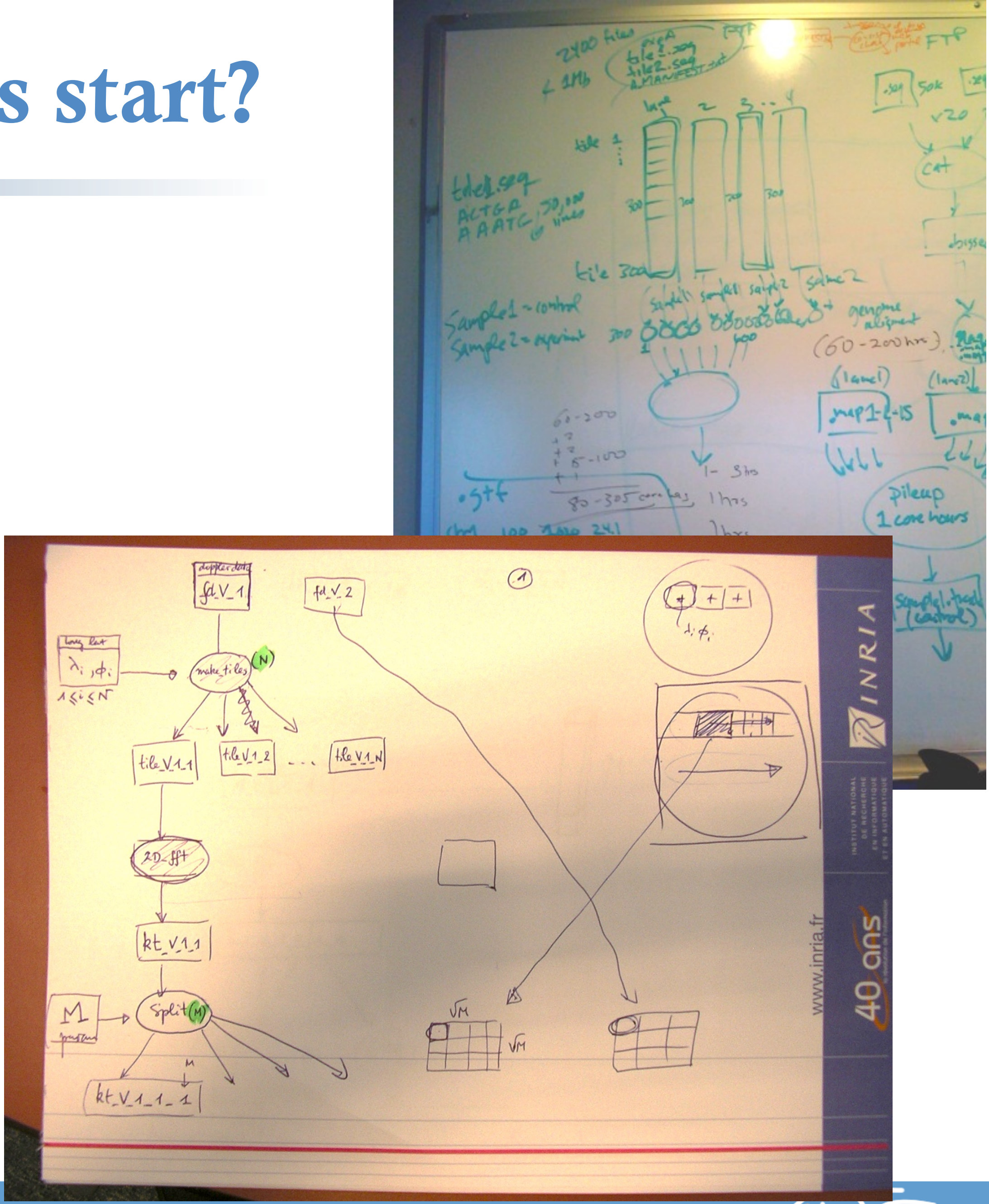


Abstract Workflow

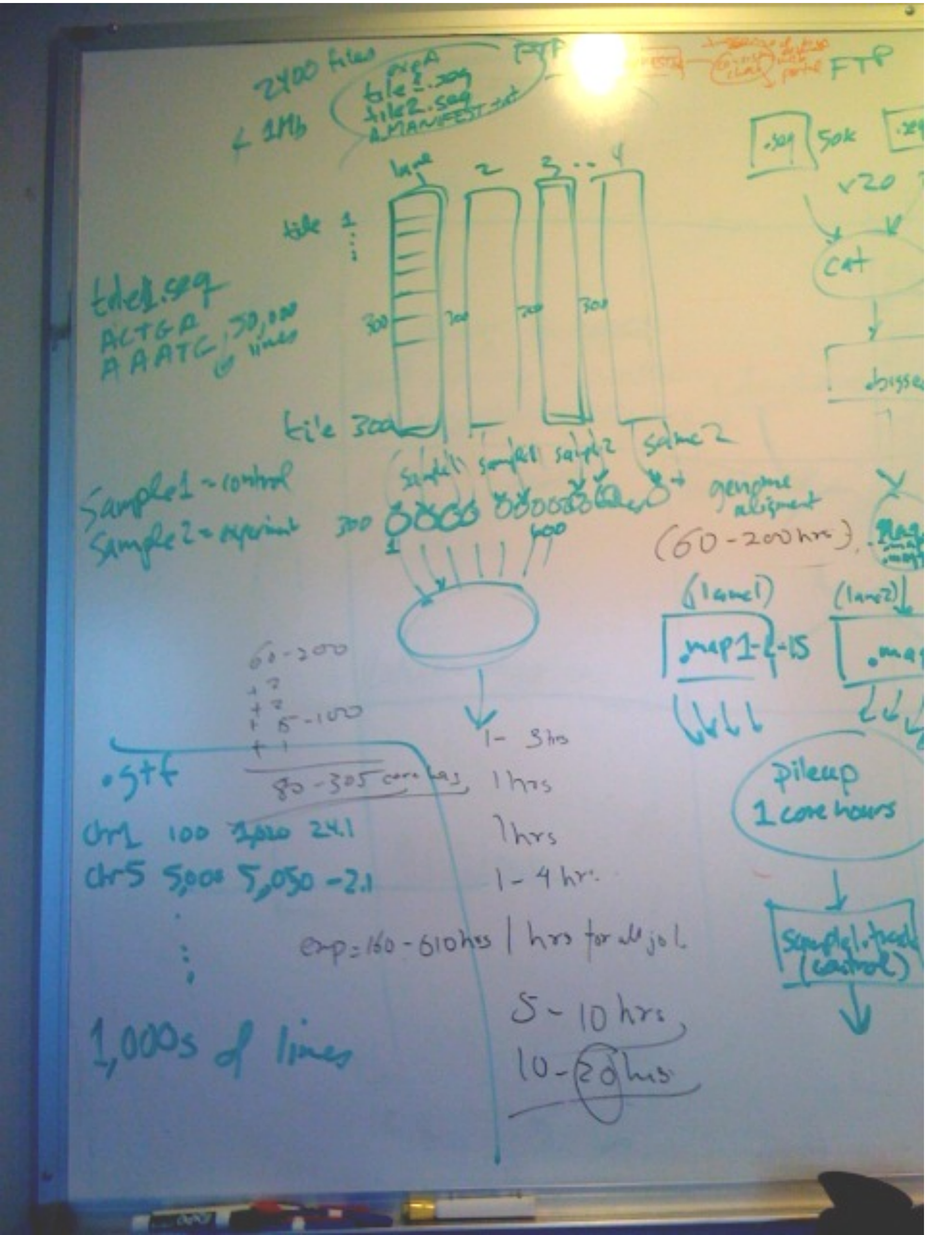
Executable Workflow

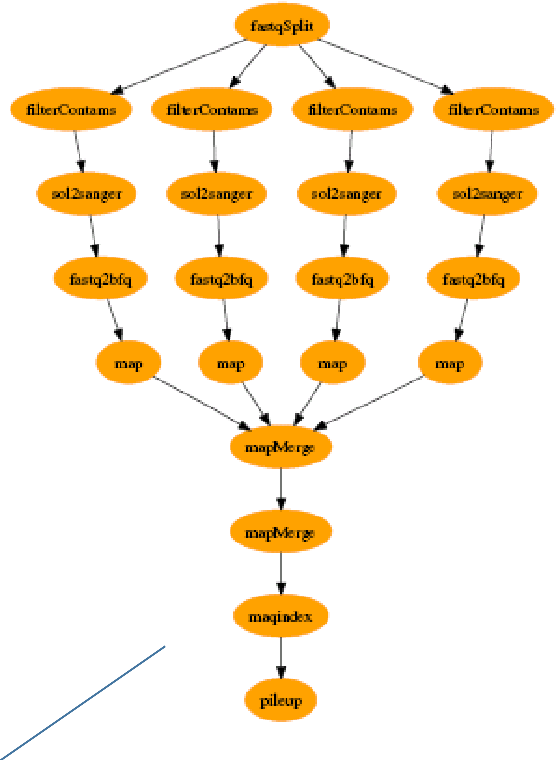
How do workflows start?



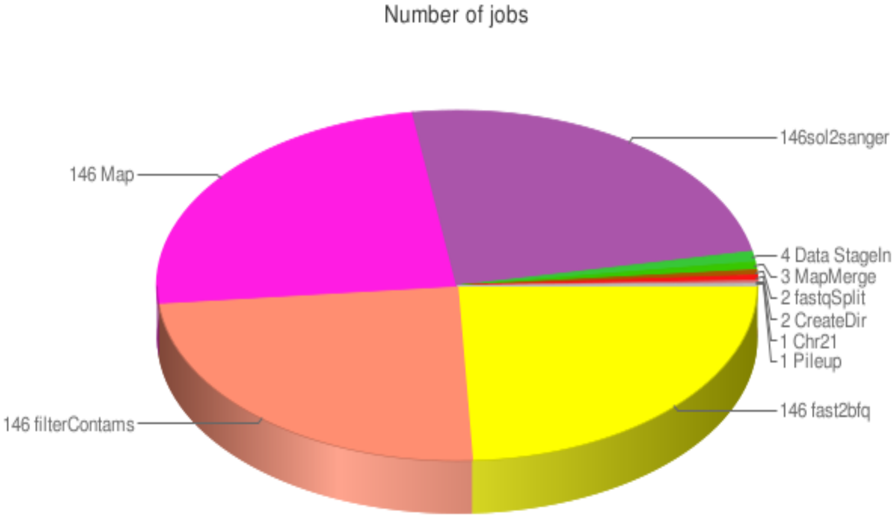


Time to solution: 2 weeks- 3 months





Execution on USC resources

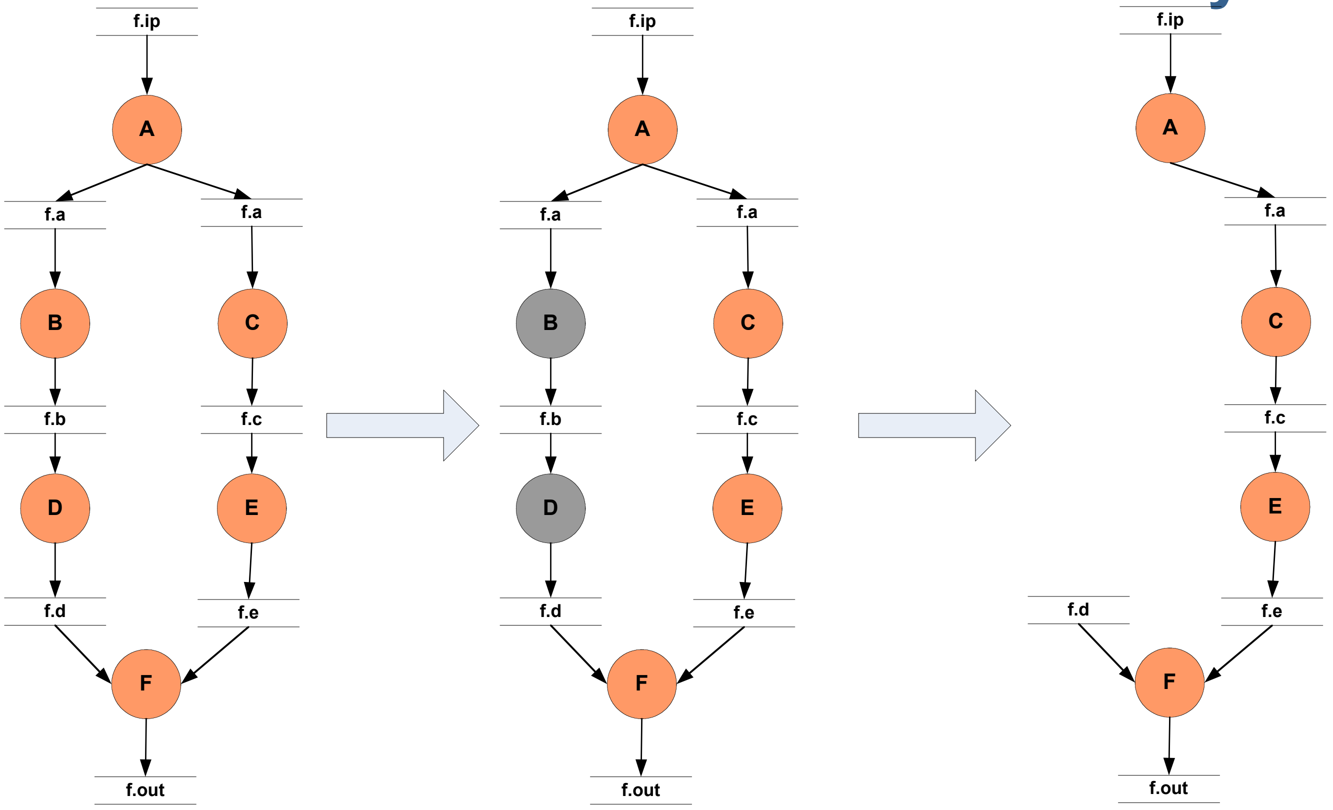


Pegasus optimizations address issues of:

- Failures in the execution environment or application
- Data storage limitations on execution sites
- Performance
  - Small workflow tasks
- Heterogeneous execution architectures
  - Different file systems (shared/non-shared)
  - Different system architectures (Cray XT, Blue Gene, …)

Sometimes fatal errors occur during workflow execution

Want to restart the workflow from where it left off
Sometimes intermediate data is already available



Abstract Workflow

File f.d exists somewhere.
Reuse it.
Mark Jobs D and B to delete

Delete Job D and Job B

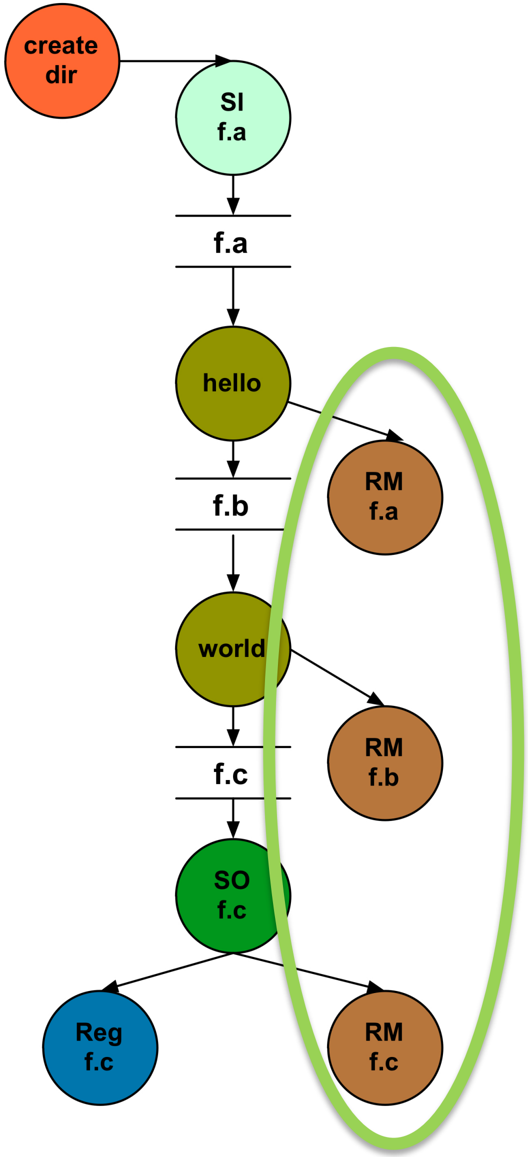
Workflow Reduction

Data Reuse

Workflow-level checkpointing

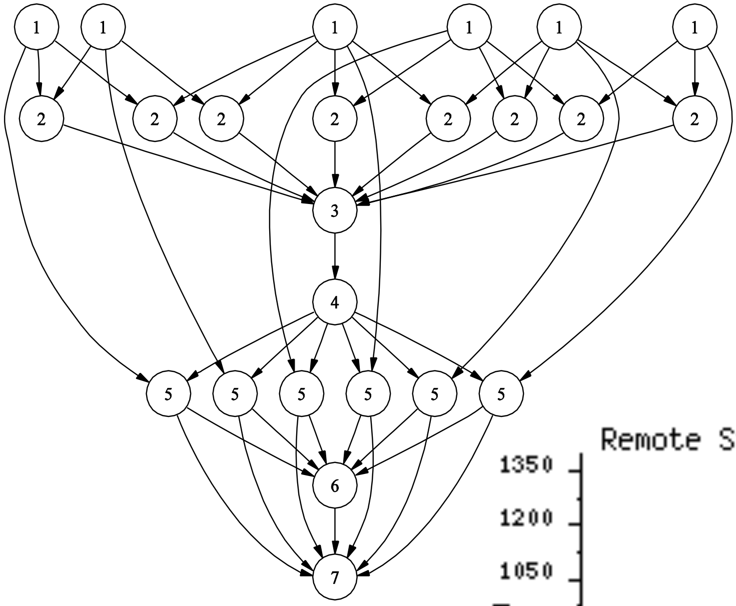
Storage limitations

“Small” amount of space

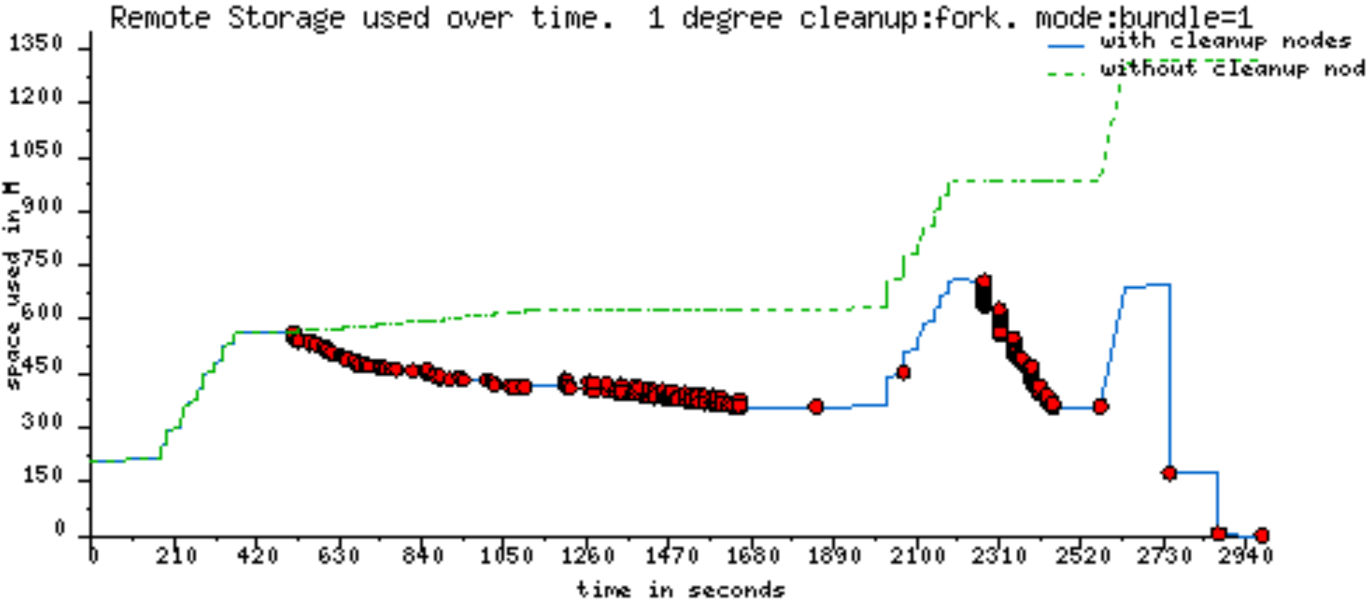


Automatically add tasks to “clean up” data no longer needed

Montage



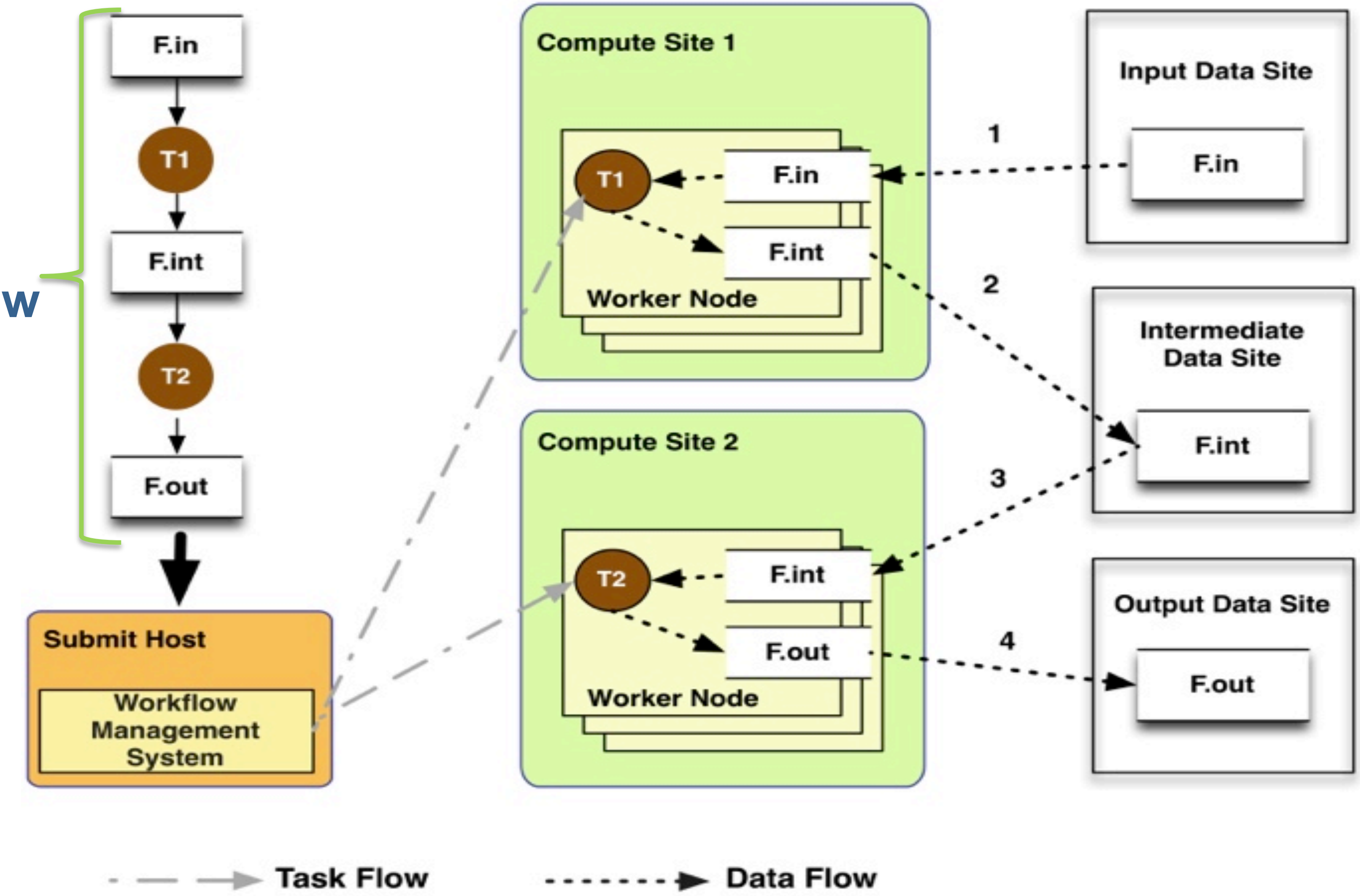
1.25GB versus 4.5 GB



Storage limitations

Variety of file system deployments: shared vs non-shared

User workflow



pegasus-transfer subsystem

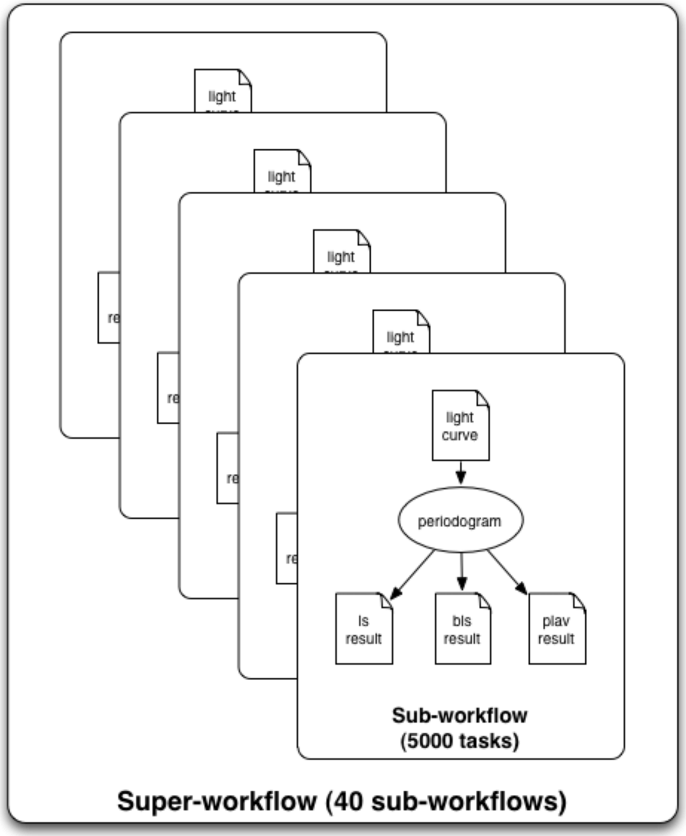
- Command line tool used internally by Pegasus workflows
- Input is a list of source and destination URLs
- Transfers the data by calling out to tools – provided by the system (cp, wget, …) Pegasus (pegasus-gridftp, pegasus-s3) or third party (gsutil)
- Transfers are parallelized
- Transfers between non-compatible protocols are split up into two transfers using the local filesystem as a staging point
  - for example: GridFTP->GS becomes GridFTP->File and File->GS

Supported URLs

GridFTP
SRM
iRods
S3
GS
SCP
HTTP
File
Symlink

Sometimes the environment is just not exactly right

Single core workload

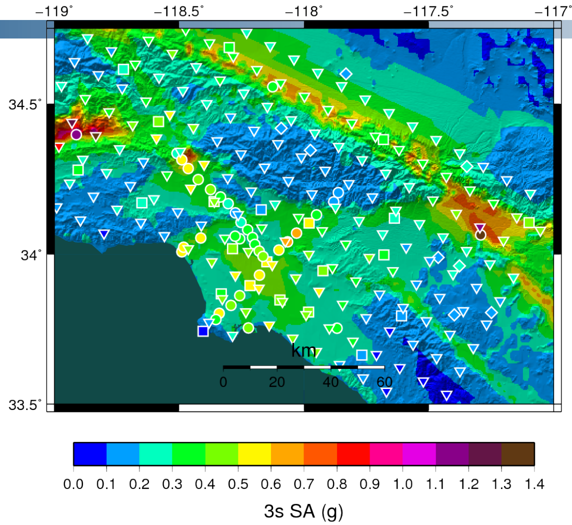


XSEDE HPC Resources

https://www.tacc.utexas.edu/resources/hpc

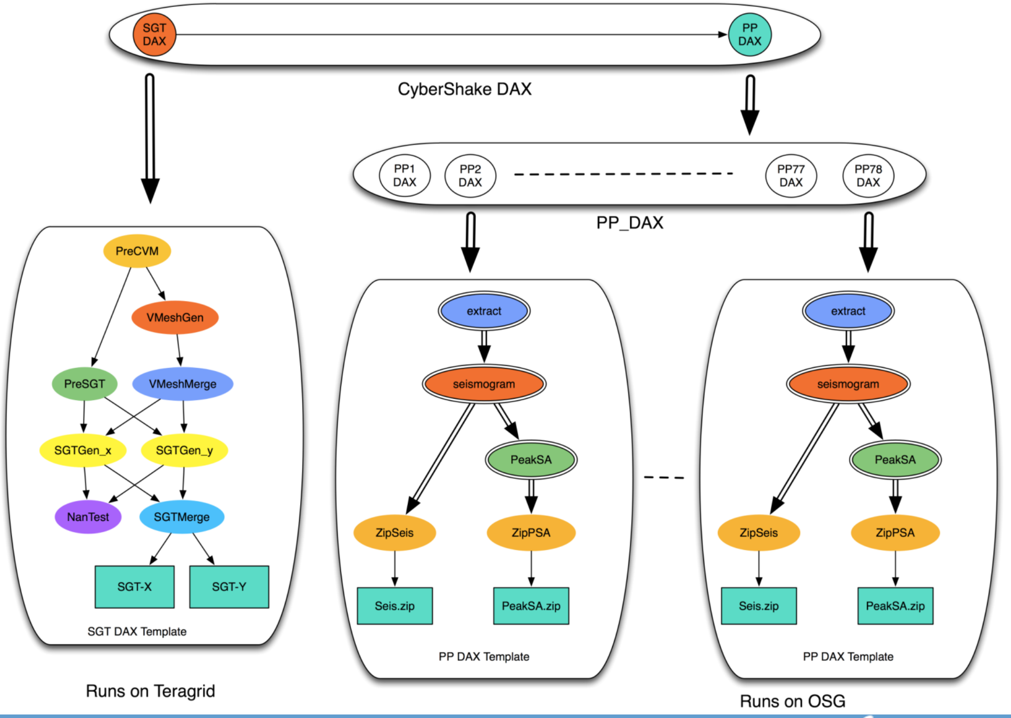
Designed for MPI codes

Southern California Earthquake Center



CyberShake PSHA Workflow

- **Description**
  - Builders ask seismologists: "What will the peak ground motion be at my new building in the next 50 years?"
  - Seismologists answer this question using Probabilistic Seismic Hazard Analysis (PSHA)



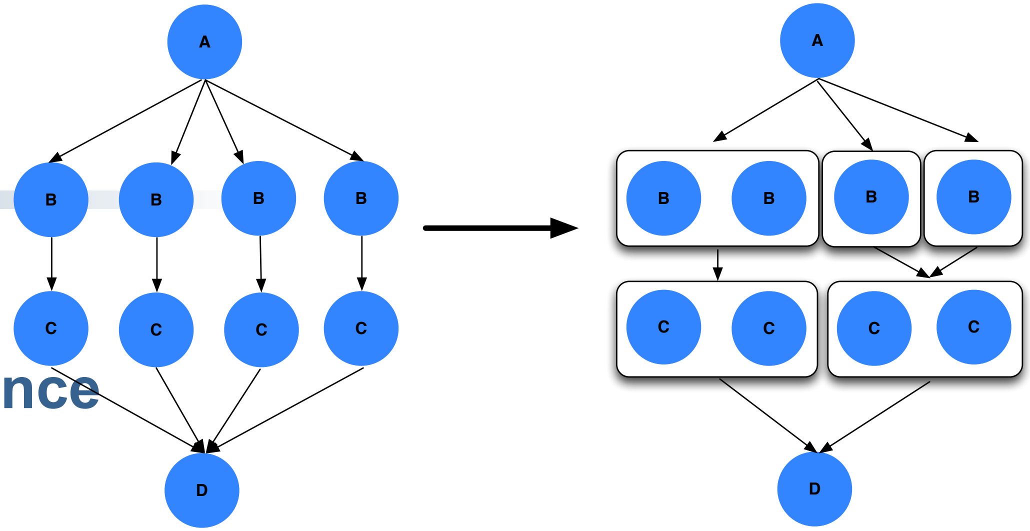
239 Workflows

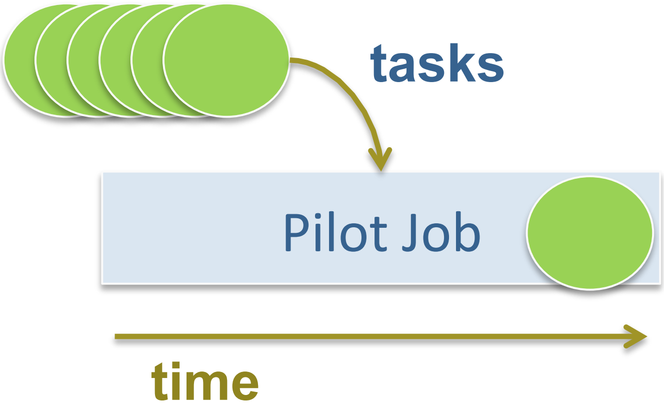
- **Each site in the input map corresponds to one workflow**
- **Each workflow has:**
  - **820,000 tasks**

Solutions

Cluster tasks

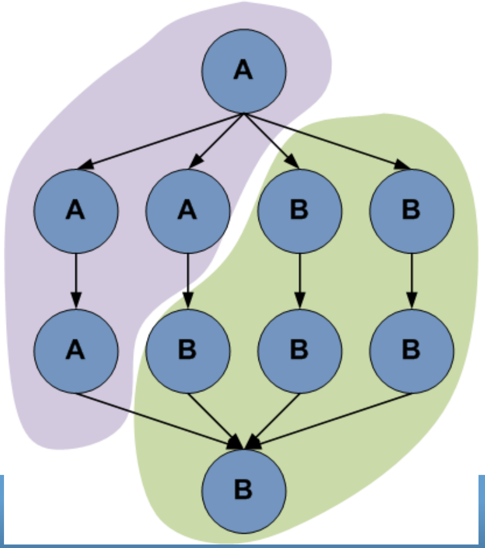
-- also good for performance





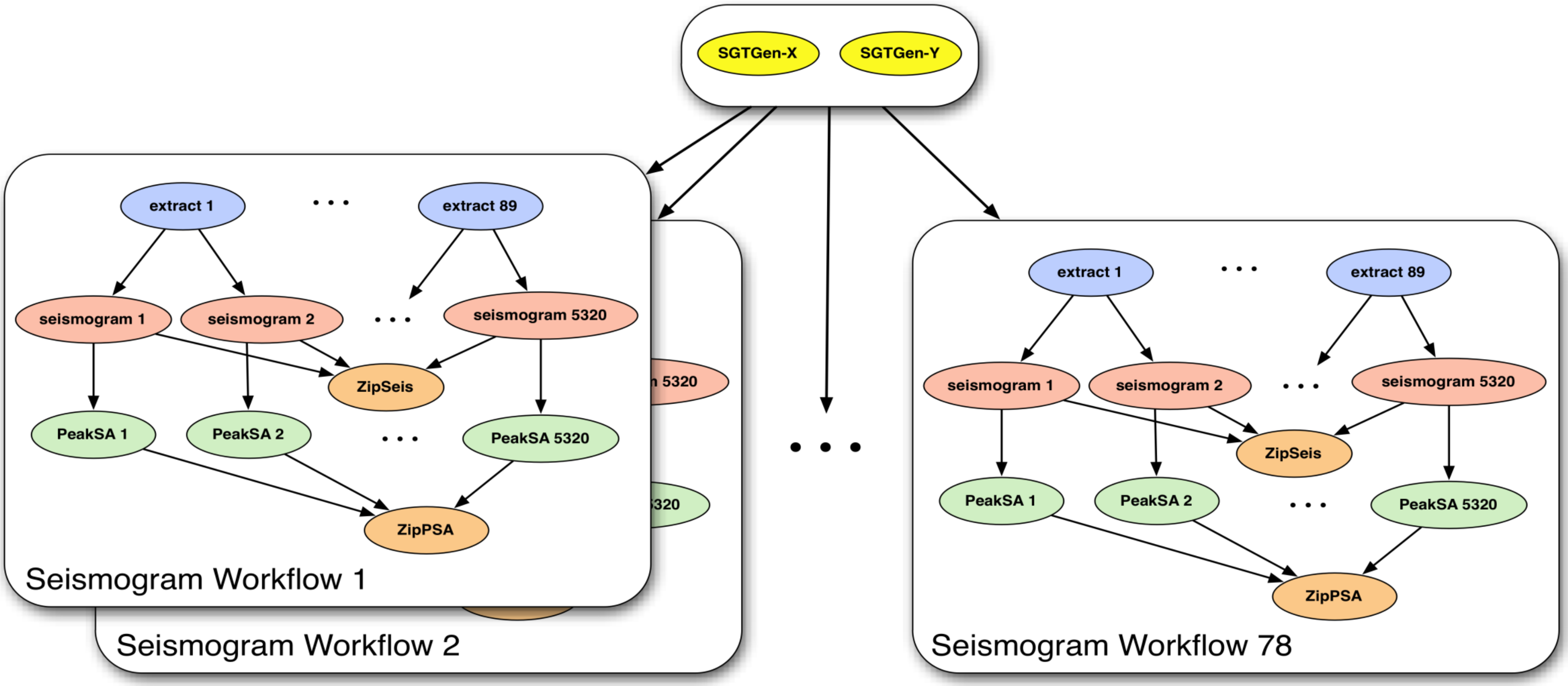
Use “pilot” jobs to dynamically provision a number of resources at a time

Use an MPI-based workflow management engine to manage sub-workflows

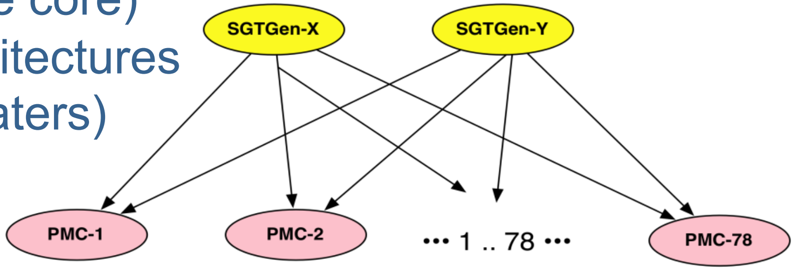


Pegasus-MPI-Cluster

- A master/worker task scheduler for running fine-grained workflows on batch systems
- Runs as an MPI job
  - Uses MPI to implement master/worker protocol
- Works on most HPC systems, used on XSEDE
  - Requires: MPI, a shared file system, and fork()
- Allows sub-graphs of a Pegasus workflow to be submitted as monolithic grid jobs to remote resources

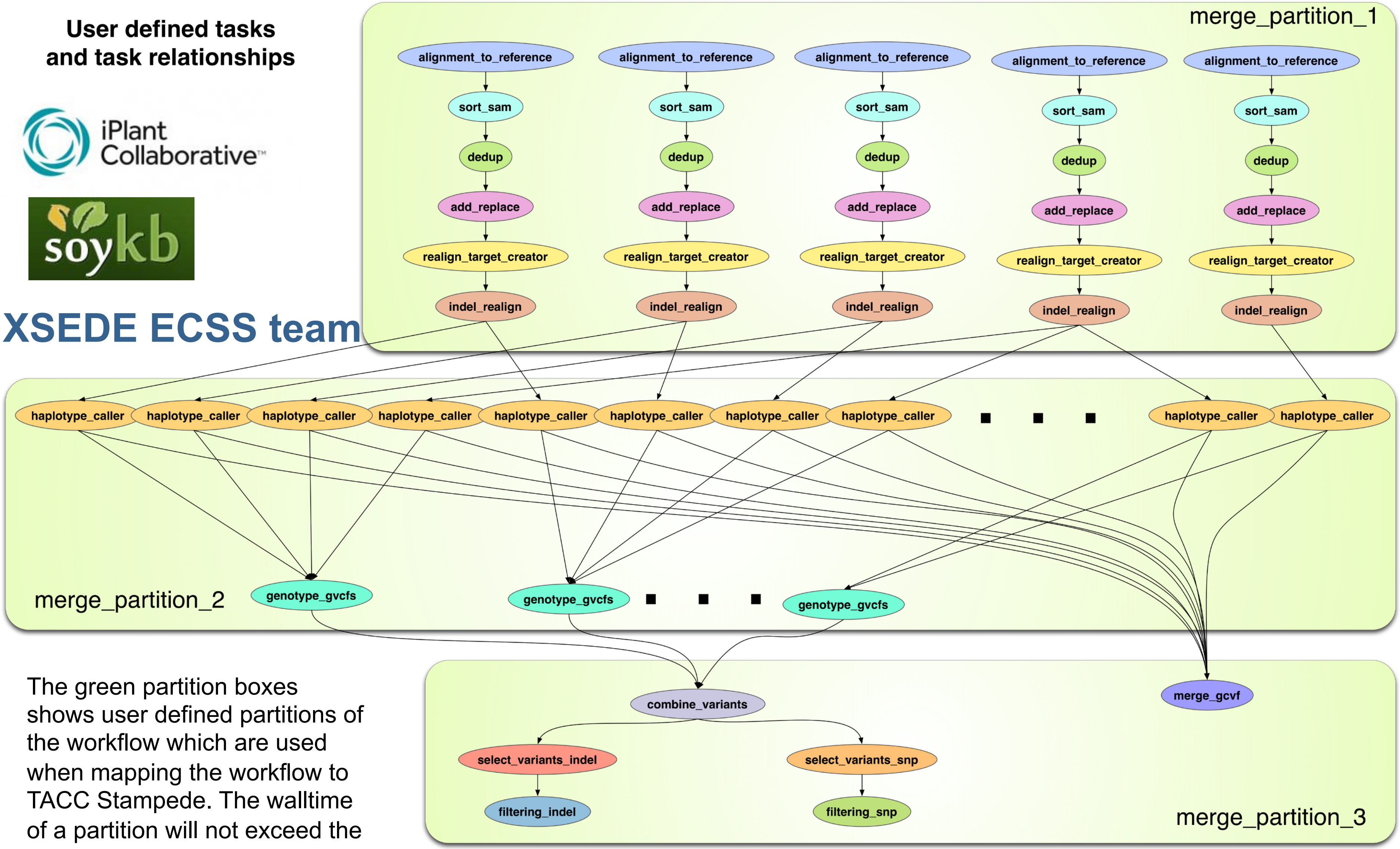


Enables earthquake scientists (SCEC) to run post-processing (single core) computations on new architectures (Stampede, Titan, Blue Waters)



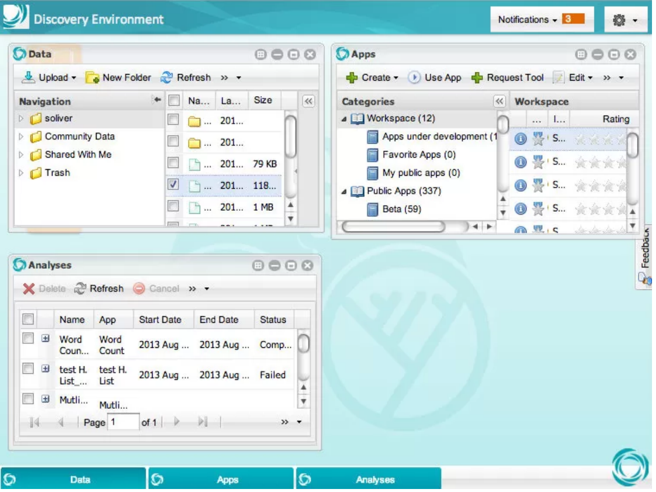
User defined tasks and task relationships

XSEDE ECSS team



The green partition boxes shows user defined partitions of the workflow which are used when mapping the workflow to TACC Stampede. The walltime of a partition will not exceed the 48 hour wall clock limit, given a certain number of compute nodes.

SoyKB: Bioinformatics analysis of 1000+ resequenced soybean germplasm lines selected for major traits including oil, protein, soybean cyst nematode resistance (SCN), abiotic stress resistance (drought, heat and salt) and root system architecture.



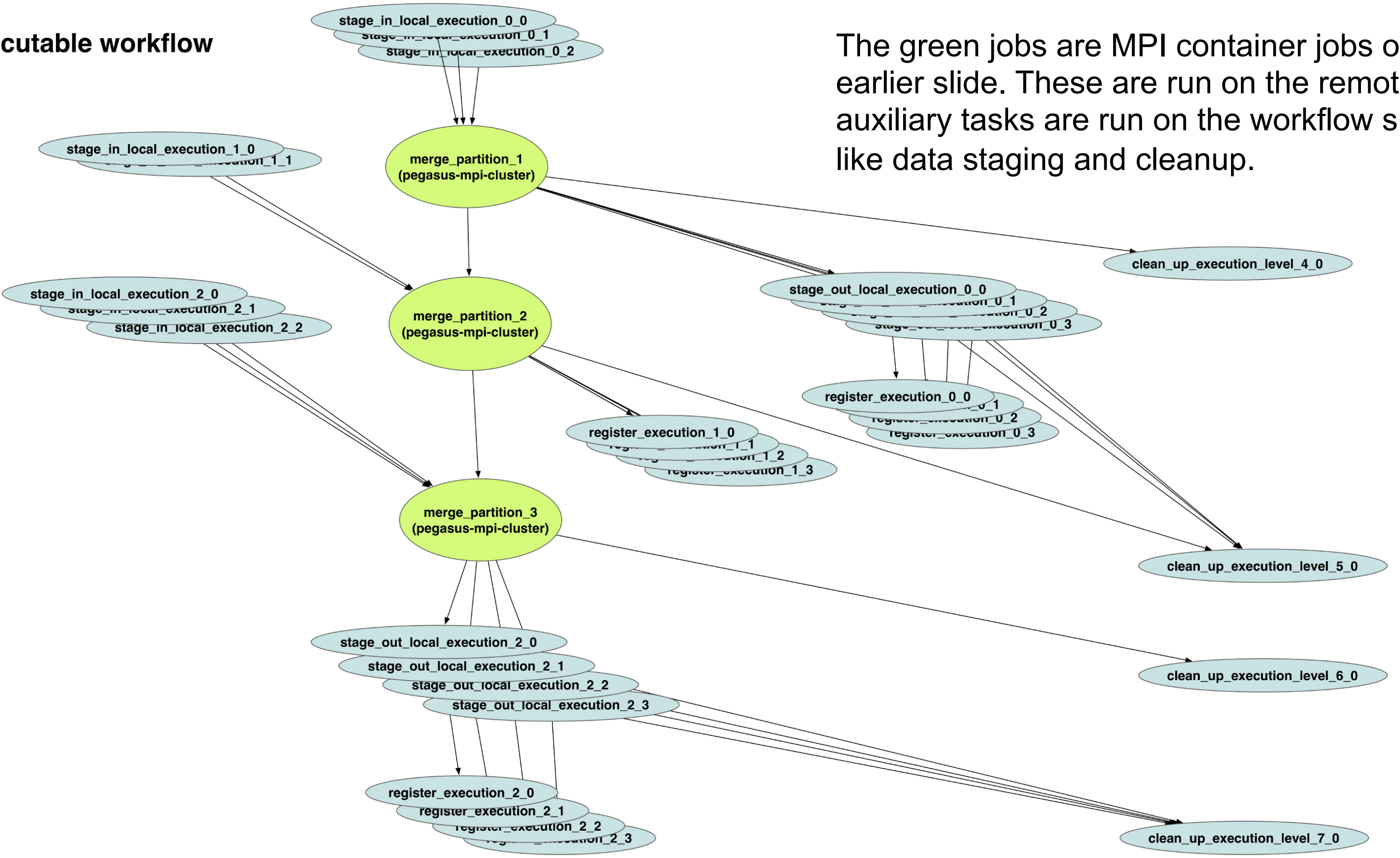
□	Name	App	Start Date	End Date	Status
□	Word Coun...	Word Count	2013 Aug ...	2013 Aug ...	Comp...
□	test H. List_...	test H. List	2013 Aug ...	2013 Aug ...	Failed
□	Mutli...	Mutli...			

Input data is fetched from iPlant Data Store, or if already replicated, from the TACC iPlant Data Store node for close to computation access

Outputs are automatically put back into the Data Store for easy access and further analysis in the iPlant Discovery Environment

Executable workflow

The green jobs are MPI container jobs of the partitions shown in the earlier slide. These are run on the remote supercomputer. The blue auxiliary tasks are run on the workflow submit host, and handle things like data staging and cleanup.


stage_in_local_execution_0_0
stage_in_local_execution_0_1
stage_in_local_execution_0_2
stage_in_local_execution_1_0
stage_in_local_execution_1_1
merge_partition_1
(pegasus-mpi-cluster)
stage_in_local_execution_2_0
stage_in_local_execution_2_1
stage_in_local_execution_2_2
merge_partition_2
(pegasus-mpi-cluster)
clean_up_execution_level_4_0
stage_out_local_execution_0_0
stage_out_local_execution_0_1
stage_out_local_execution_0_2
stage_out_local_execution_0_3
register_execution_0_0
register_execution_0_1
register_execution_0_2
register_execution_0_3
register_execution_1_0
register_execution_1_1
register_execution_1_2
register_execution_1_3
merge_partition_3
(pegasus-mpi-cluster)
clean_up_execution_level_5_0
stage_out_local_execution_2_0
stage_out_local_execution_2_1
stage_out_local_execution_2_2
stage_out_local_execution_2_3
clean_up_execution_level_6_0
register_execution_2_0
register_execution_2_1
register_execution_2_2
register_execution_2_3
clean_up_execution_level_7_0

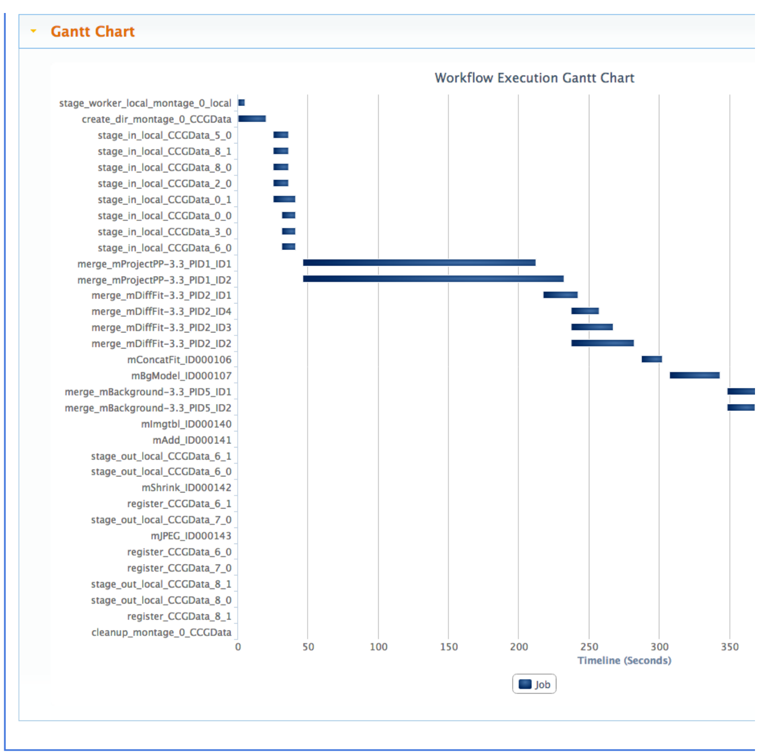
iPlant Collaborative™

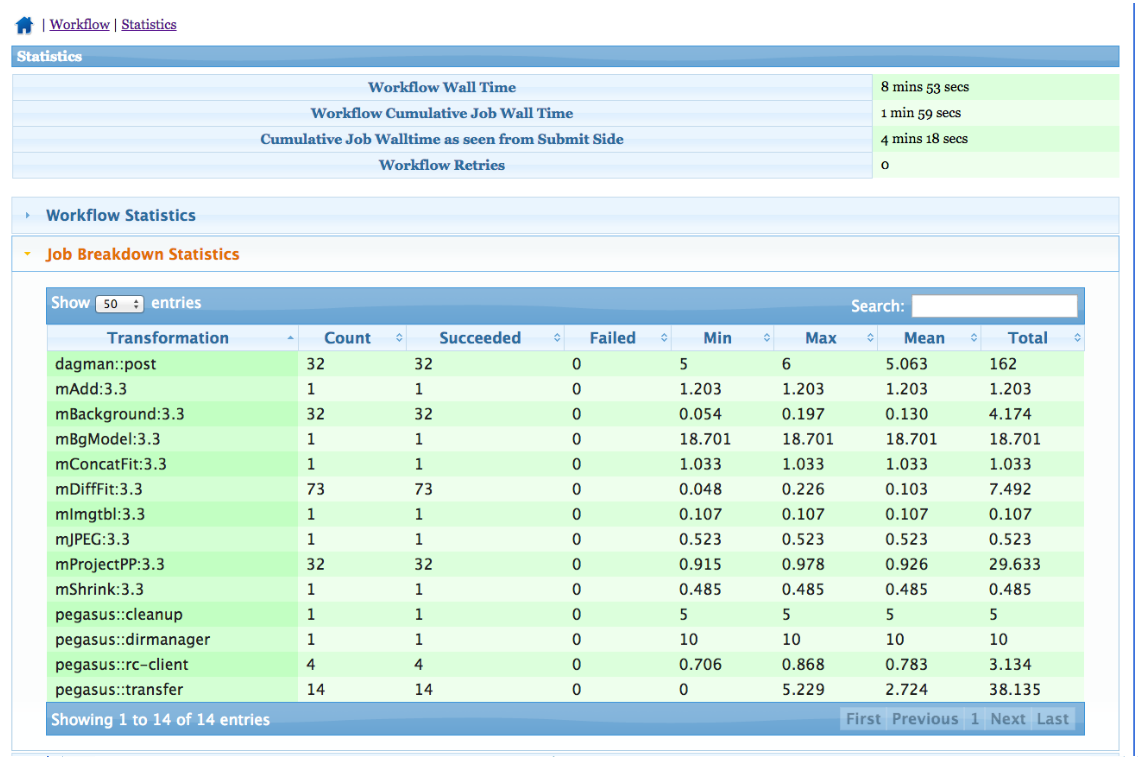
soykb

Pegasus-kickstart

- Lightweight C based executable to launch jobs
- Captures job runtime provenance and logs it as a XML record
- Following information is captured about each job on all supported platforms
  - exit code with which the job it launched exited
  - start time and duration of the job
  - hostname and IP address of the host the job ran on
  - stdout and stderr of the job
  - arguments with which it launched the job
  - directory in which the job was launched
  - environment that was set for the job before it was launched
- Additional profiling
  - peak memory usage (resident set size, and vm size)
  - total I/O read and write,
  - Pid
  - all files accessed (total read and write per file)

Workflow Monitoring Dashboard – *pegasus-dashboard*





Statistics

Workflow Wall Time	8 mins 53 secs
Workflow Cumulative Job Wall Time	1 min 59 secs
Cumulative Job Walltime as seen from Submit Side	4 mins 18 secs
Workflow Retries	0

Workflow Statistics

Job Breakdown Statistics

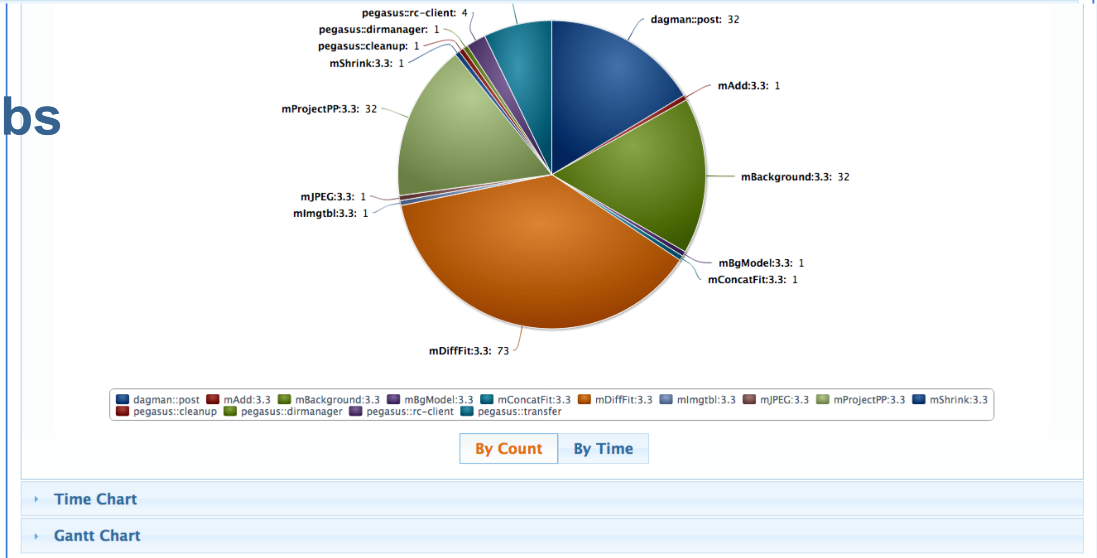
Show 50 entries   Search:

Transformation	Count	Succeeded	Failed	Min	Max	Mean	Total
dagman::post	32	32	0	5	6	5.063	162
mAdd:3.3	1	1	0	1.203	1.203	1.203	1.203
mBackground:3.3	32	32	0	0.054	0.197	0.130	4.174
mBgModel:3.3	1	1	0	18.701	18.701	18.701	18.701
mConcatFit:3.3	1	1	0	1.033	1.033	1.033	1.033
mDiffFit:3.3	73	73	0	0.048	0.226	0.103	7.492
mImgtbl:3.3	1	1	0	0.107	0.107	0.107	0.107
mJPEG:3.3	1	1	0	0.523	0.523	0.523	0.523
mProjectPP:3.3	32	32	0	0.915	0.978	0.926	29.633
mShrink:3.3	1	1	0	0.485	0.485	0.485	0.485
pegasus::cleanup	1	1	0	5	5	5	5
pegasus::dirmanager	1	1	0	10	10	10	10
pegasus::rc-client	4	4	0	0.706	0.868	0.783	3.134
pegasus::transfer	14	14	0	0	5.229	2.724	38.135

Showing 1 to 14 of 14 entries

Status, statistics, timeline of jobs

Helps pinpoint errors



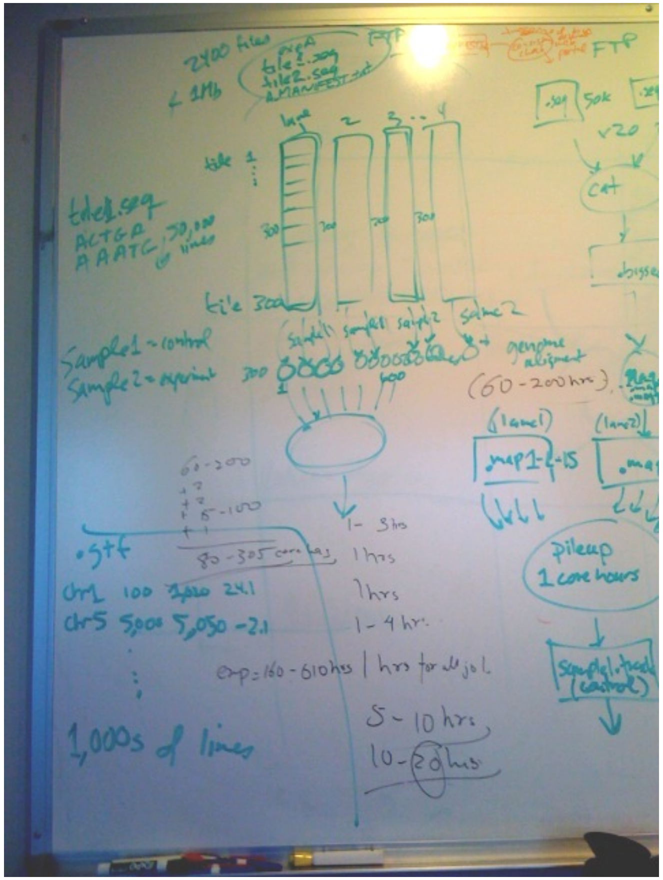
If you are interested in Pegasus

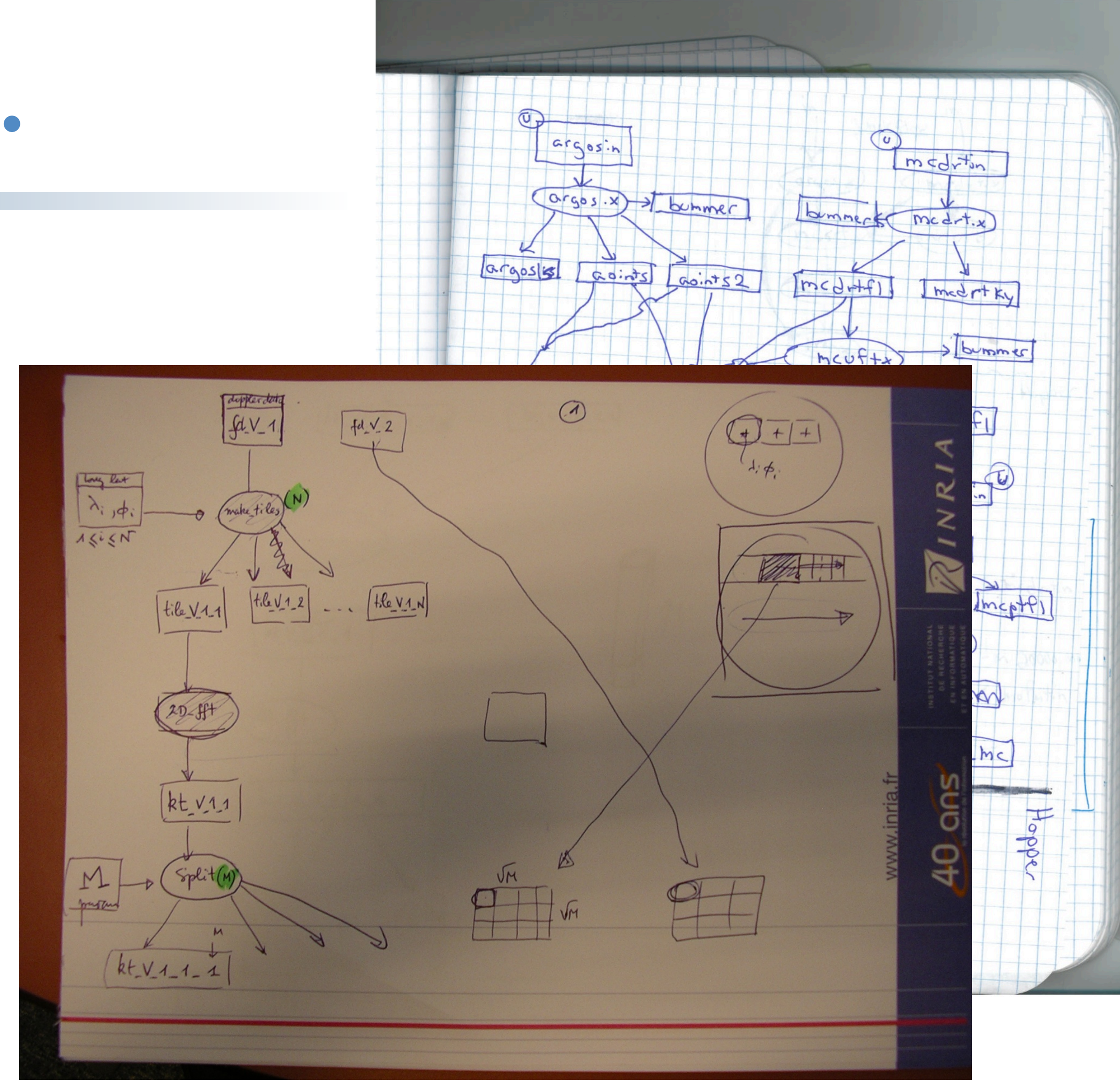
- Pegasus: http://pegasus.isi.edu
- Tutorial and documentation: http://pegasus.isi.edu/wms/docs/latest/
- Virtual Machine with all software and examples
  http://pegasus.isi.edu/downloads
- Take look at some Pegasus applications:

  http://pegasus.isi.edu/applications
- User Support available: pegasus-users@isi.edu

If you get stuck…

And you can draw….





The XSEDE ECSS and Pegasus teams can help you!