Implications of Data-Intensive Analysis on Scientific Workflow Management

Ewa Deelman, Ph.D.

Science Automation Technologies Group

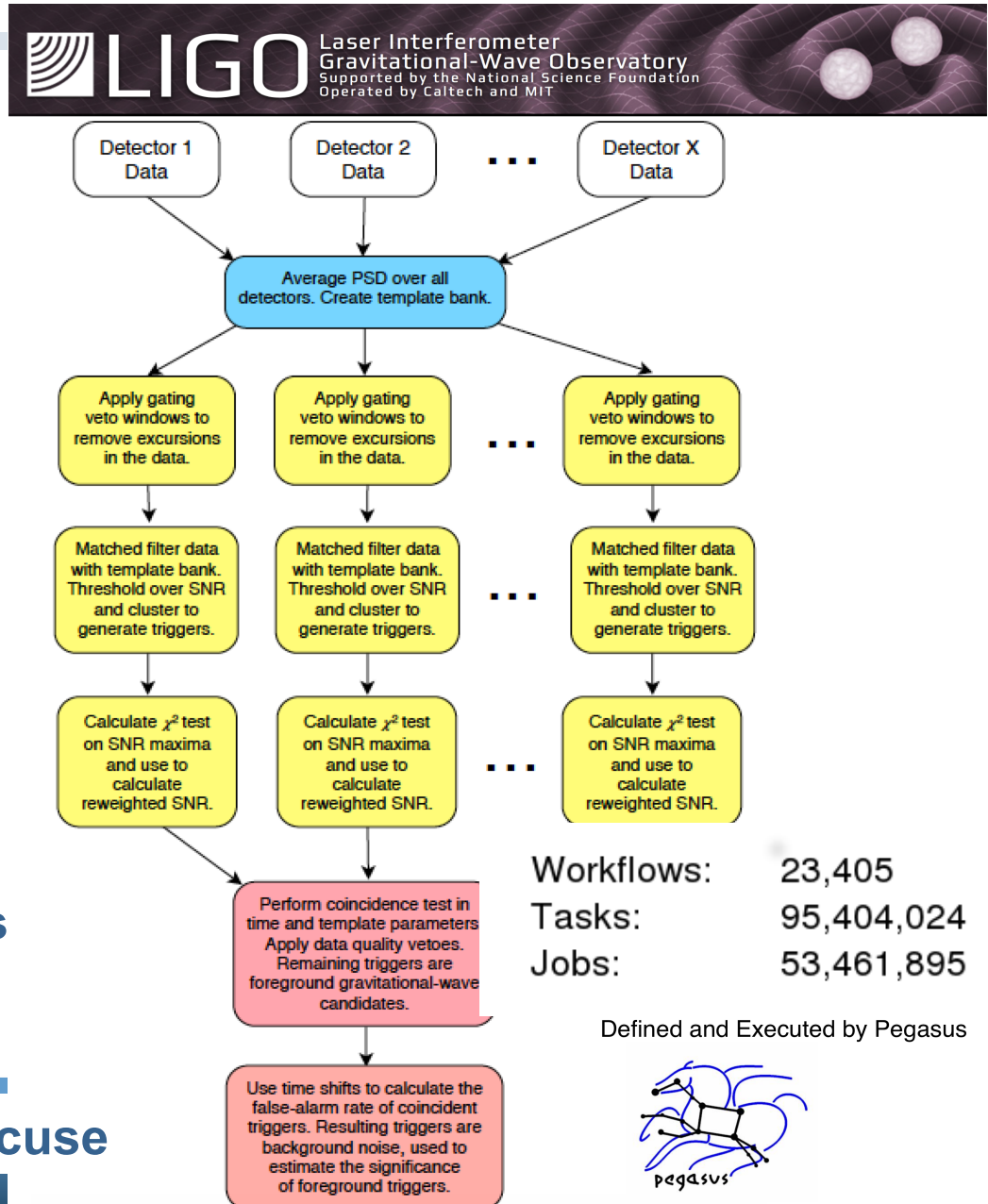
USC Information Sciences Institute

Funding from DOE, NSF, and NIH

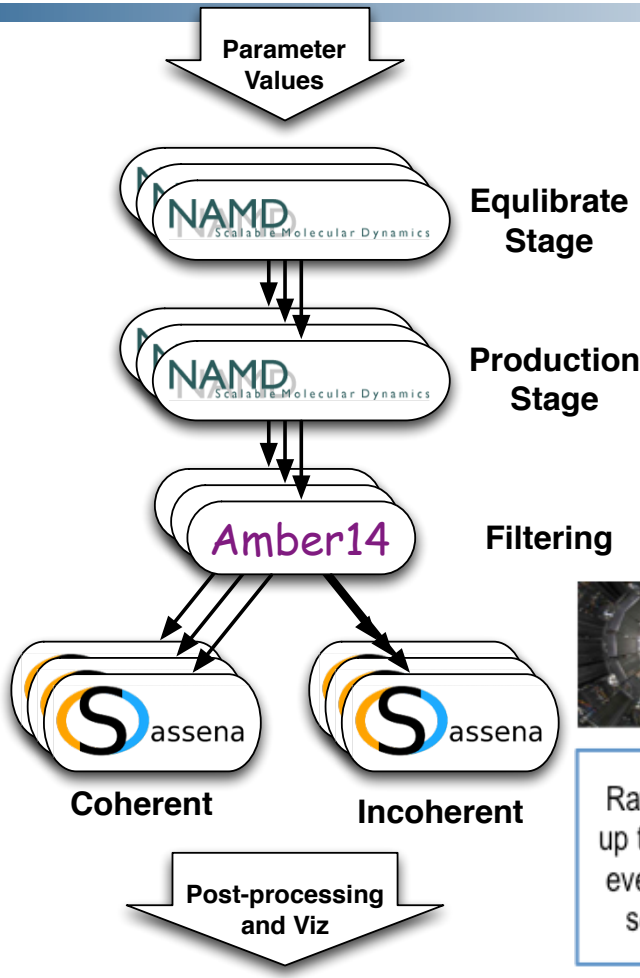
Smoky Mountains Computational Science and Engineering Conference 2016

LIGO's Gravitational Wave detection high-throughput workloads

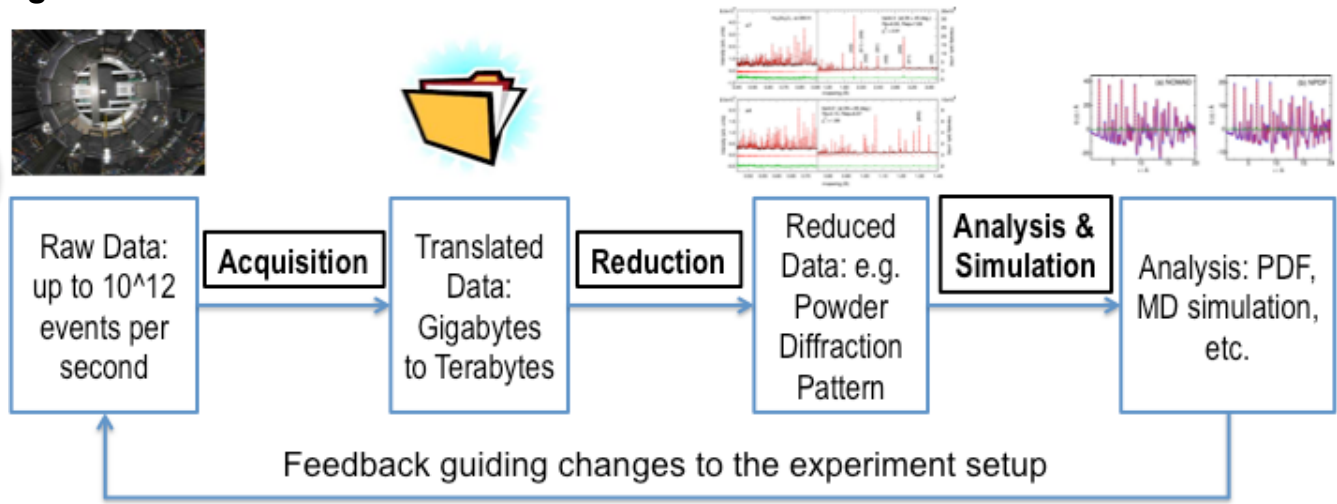
- One of the main pipelines to measure the statistical significance of data needed for discovery
- Contains 100's of thousands of jobs and accesses on order of terabytes of data
- Uses data from multiple detectors
- For the detection, the pipeline was executed on Syracuse and Albert Einstein Institute Hannover
- Use our Pegasus software to automate the execution of tasks and data access



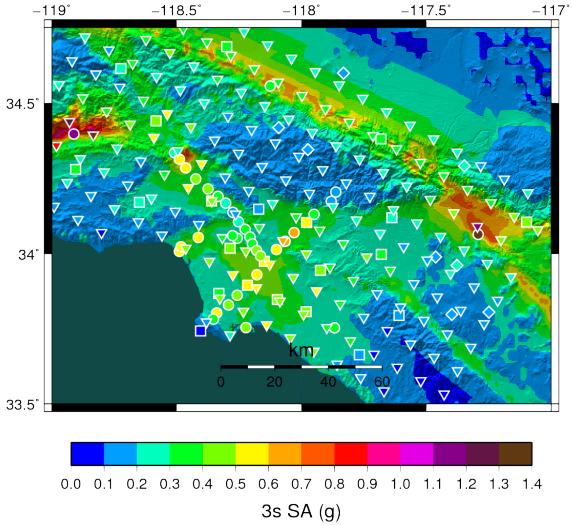
Molecular Dynamics Modeling: Tightly Coupled Parallel Workloads



SNS refinement workflow executes a parameter sweep of molecular dynamics and neutron scattering simulations to optimize the value for a target parameter to the experimental data

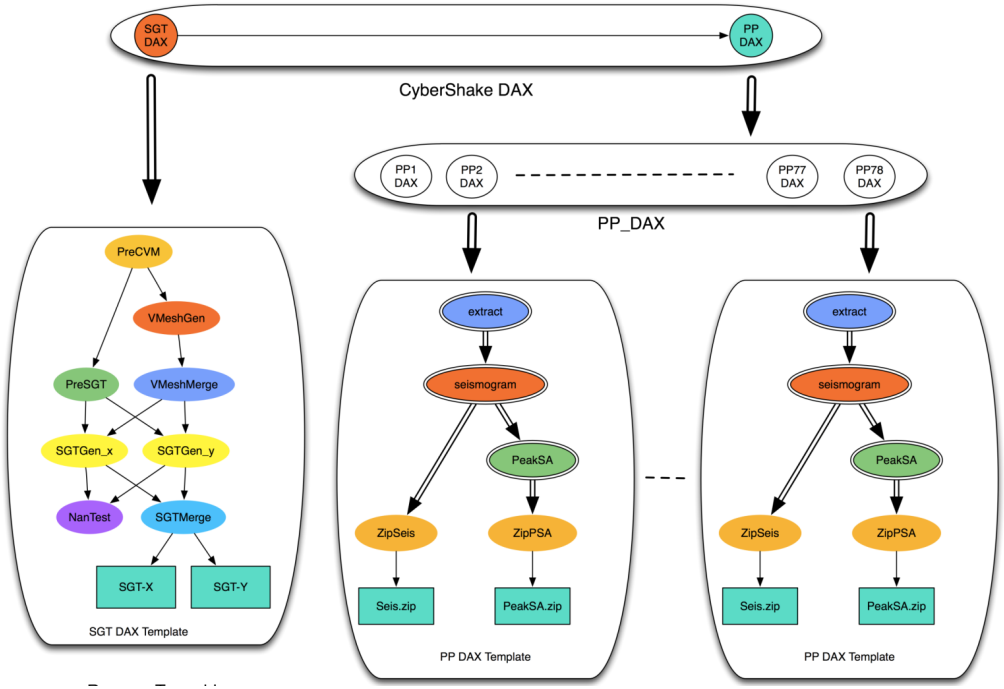


Southern California Earthquake Center, Heterogeneous workloads



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)

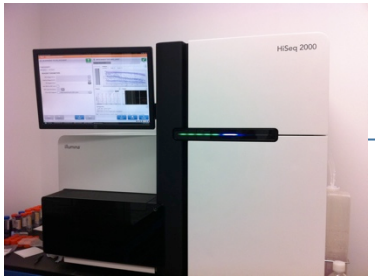


239 Workflows

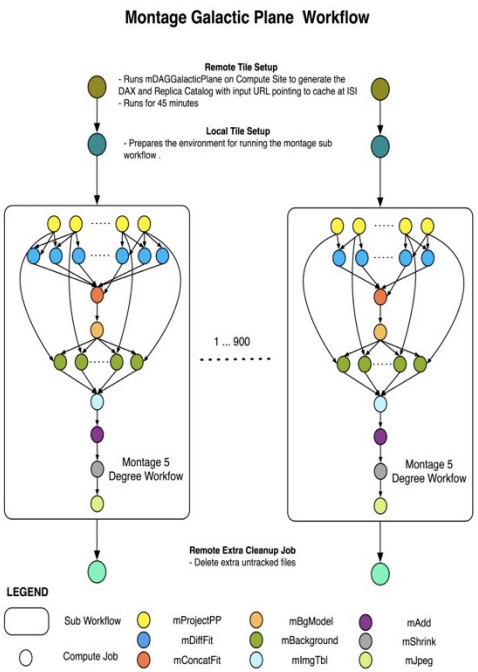
- **Each site in the input map corresponds to one workflow**
- **Each workflow has:**
  - **820,000 tasks**

Scott Callaghan, Phil Maechling, USC

Sometimes the environment is complex



Data Storage



Work definition

Local Resource

Campus Cluster

XSEDE

ALCF

NERSC

OLCF

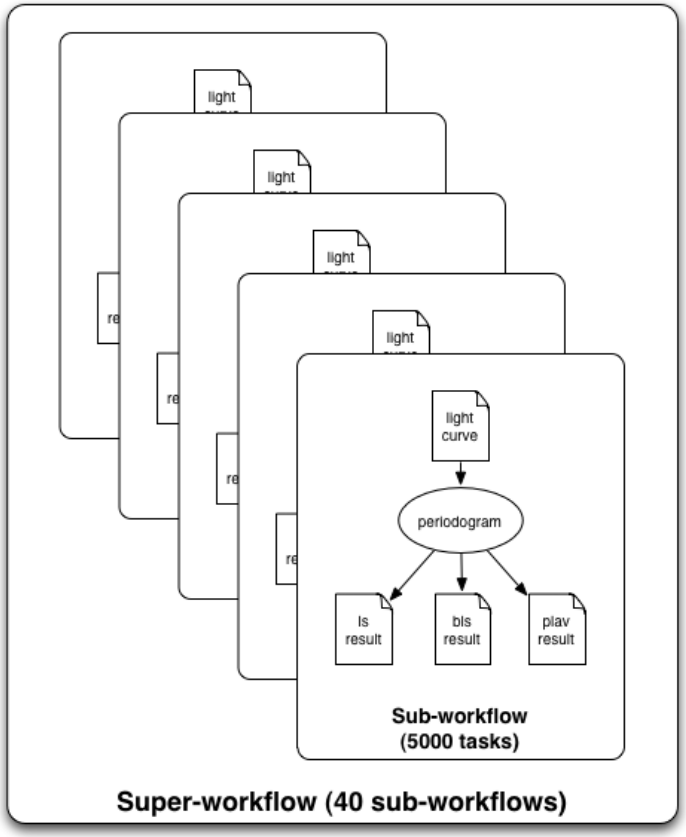
Open Science Grid

Chameleon

Amazon Cloud

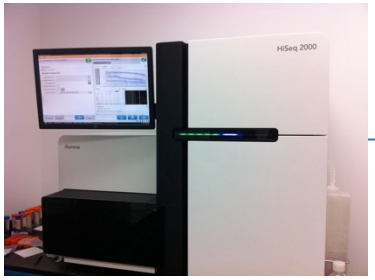
Sometimes the environment is just not exactly right

Single core workload



Cray XK7 System Environment / Designed for MPI codes

Our Approach: Submit locally, Compute globally



Data Storage

data

Campus Cluster

XSEDE

NERSC

ALCF

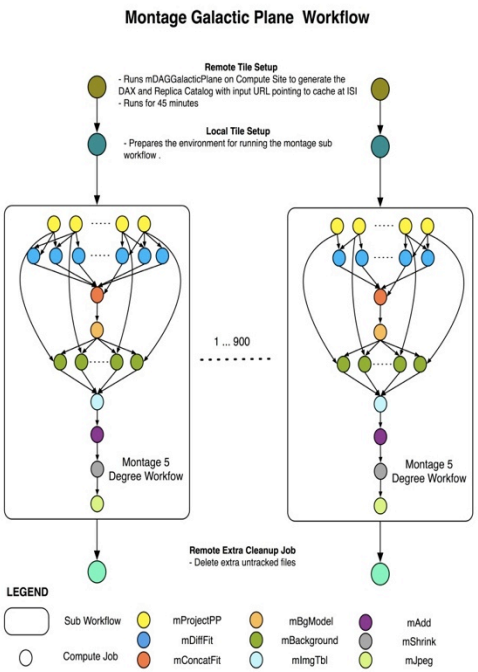
OLCF

Open Science Grid

Chameleon

Amazon Cloud

work



Work definition

Workflow Management System

Local Resource

Workflow Management System (WMS) Functions

- **Automate computational scientific methods that scientists rely on for their work**
- Discover what resources (computation, data, software) are available
- Select the appropriate resources based on a architecture, availability of software, performance, reliability, availability of cycles, storage,..
- Devise a plan:
  - What resources to use
  - How to best adapt the workflow to the resources
  - What protocols to use to access the data, to schedule jobs
  - What data to save
- Execute the plan
  - In a reliable way
  - Keep track of what data was accessed, generated and how
- Currently Outside of the WMS functions
  - Resource provisioning

Pegasus Workflow Management System (est. 2001)

Collaboration with HTCondor, UW Madison

- A workflow "compiler"/planner
  - 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*
- Web-based monitoring and monitoring

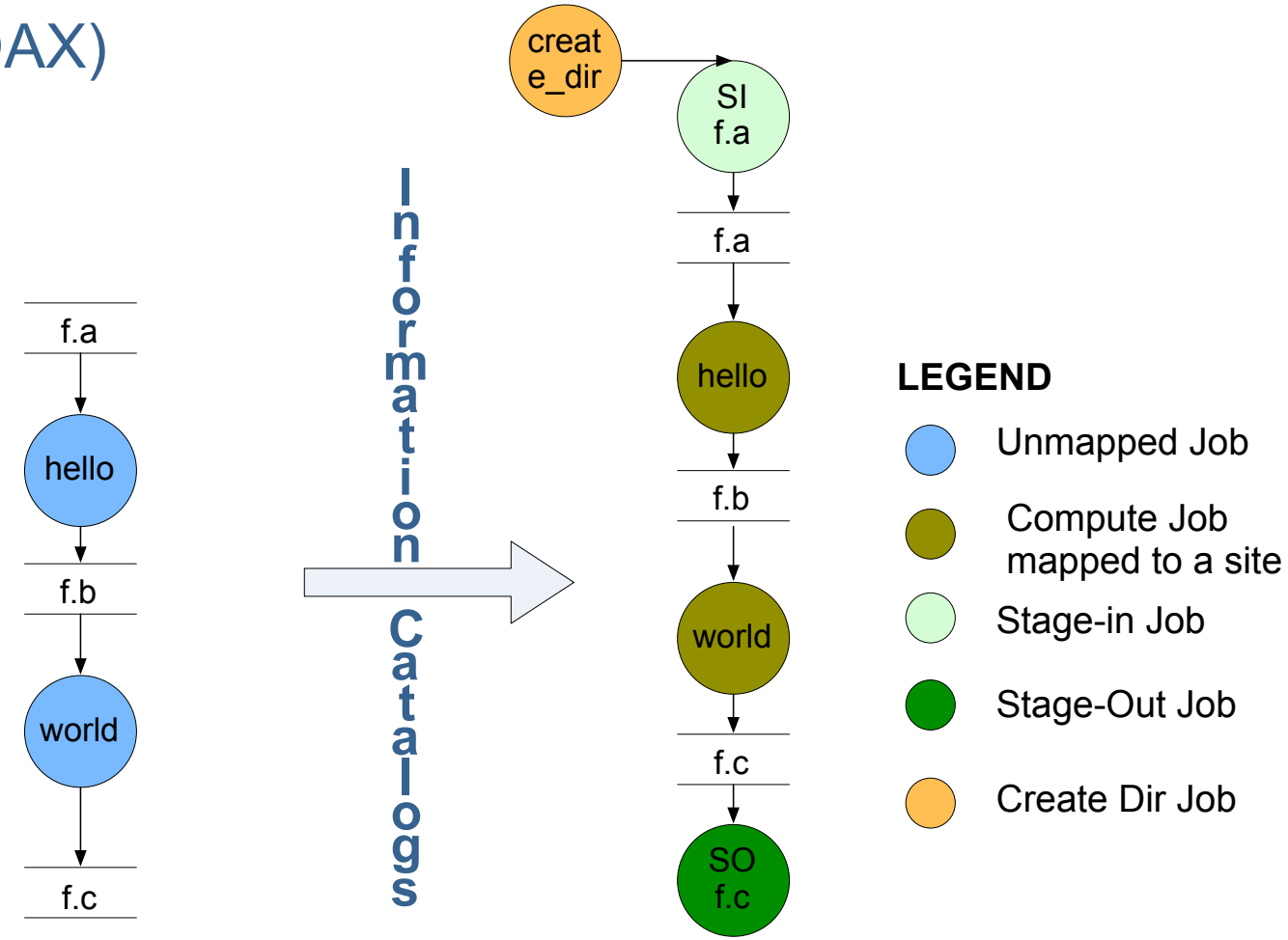
Generating executable workflows

(DAX)

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

Java, Perl, Python

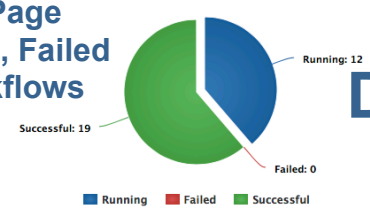
R - prototype



Pegasus Dashboard

Workflow Wall Time	5 days 1 hour
Workflow Cumulative Job Wall Time	2065 days 10 hours
Cumulative Job Walltime as seen from Submit Side	2066 days 23 hours
Workflow Cumulative Badput Time	58 mins 32 secs
Cumulative Job Badput Walltime as seen from Submit Side	1 hour 32 secs
Workflow Retries	5

Workflow Listing Page Shows Successful, Failed and Running Workflows



how results for all

Show 10 entries Search: injections

Workflow Label	Submit Host	Submit Directory	State	Submitted On
analysis2-C01-injections	sugar-dev2.phy.syr.edu	/usr1/amber.lenon/pycbc-tmp.4a07mk2LXe/work	Successful	Sat, 06 Feb 2016 13:27:15
analysis8-C01-injections	sugar-dev2.phy.syr.edu	/usr1/amber.lenon/pycbc-tmp.cqBQirspEl/work	Successful	Mon, 08 Feb 2016 15:25:05
analysis7-C01-injections	sugar-dev2.phy.syr.edu	/usr1/amber.lenon/pycbc-tmp.tpHketeH7X/work	Successful	Mon, 08 Feb 2016 11:45:22
analysis3-C01-injections	sugar-dev2.phy.syr.edu	/usr1/amber.lenon/pycbc-tmp.LU6iToRyVA/work	Running	Tue, 23 Feb 2016 16:27:30
analysis4-C01-injections	sugar-dev2.phy.syr.edu	/usr1/amber.lenon/pycbc-tmp.gArV7UNU9C/work	Running	Tue, 23 Feb 2016 16:27:44
analysis5-C01-injections	sugar-dev2.phy.syr.edu	/usr1/amber.lenon/pycbc-tmp.d95VFhwkKu/work	Running	Wed, 24 Feb 2016 11:49:17
analysis6-C01-injections	sugar-dev2.phy.syr.edu	/usr1/amber.lenon/pycbc-tmp.O1anUn5hGE/work	Running	Wed, 24 Feb 2016 11:55:55
analysis9-C01-injections	sugar-dev2.phy.syr.edu	/usr1/amber.lenon/pycbc-tmp.Lf2hB7UTuG/work	Running	Wed, 24 Feb 2016 12:07:11

Showing 11 to 18 of 18 entries (filtered from 33 total entries) First Previous 1 2 Next Last

Workflow Statistics

- Workflow Statistics
- Job Breakdown Statistics

Show 10 entries Search:

Transformation	Count	Succeeded	Failed	Min	Max	Mean	Total
dagman::post	15301	14819	482	5	554	9.607	146993
inspiral-FULL_DATA-H1_ID9	7621	7620	1	901.357	20055.080	12507.034	95316108.773
inspiral-FULL_DATA-L1_ID10	6641	6640	1	1589.662	19588.902	12504.586	83042955.049
pegasus::transfer	108	108	0	0	205.304	9.664	1043.731
coinc-FULL_DATA_FULL-H1L1_ID14	20	20	0	263.256	348.672	297.379	5947.588
pegasus::dirmanager	7	7	0	0	5	2.857	20
condor::dagman	6	6	0	607	833	700.167	4201
dagman::pre	6	6	0	11	75	27.167	163
single_template-P1_0-H1_ID5	5	5	0	360.602	383.866	373.194	1865.968
single_template_plot-P1_0-H1_ID6	5	5	0	4.013	8.382	5.008	25.041

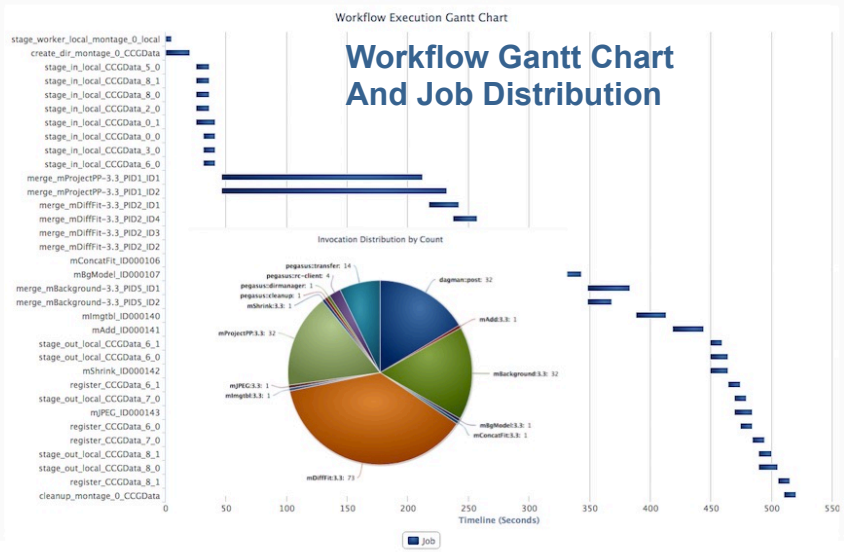
Showing 1 to 10 of 314 entries First Previous 1 2 3 4 5 32 Next Last

- Job Statistics

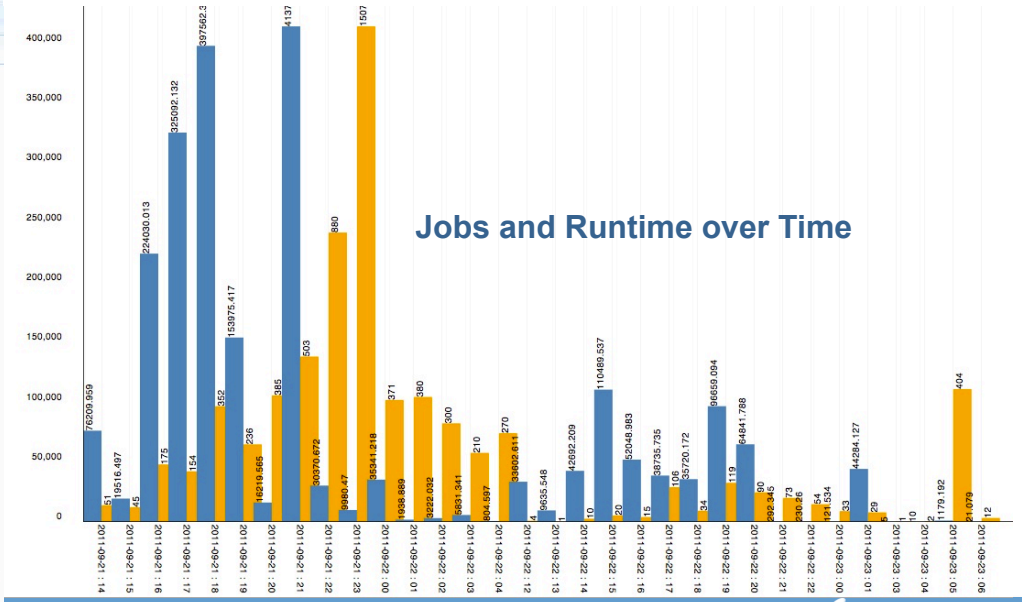
Charts

- Job Distribution
- Time Chart
- Gantt Chart

Workflow Gantt Chart And Job Distribution



Jobs and Runtime over Time

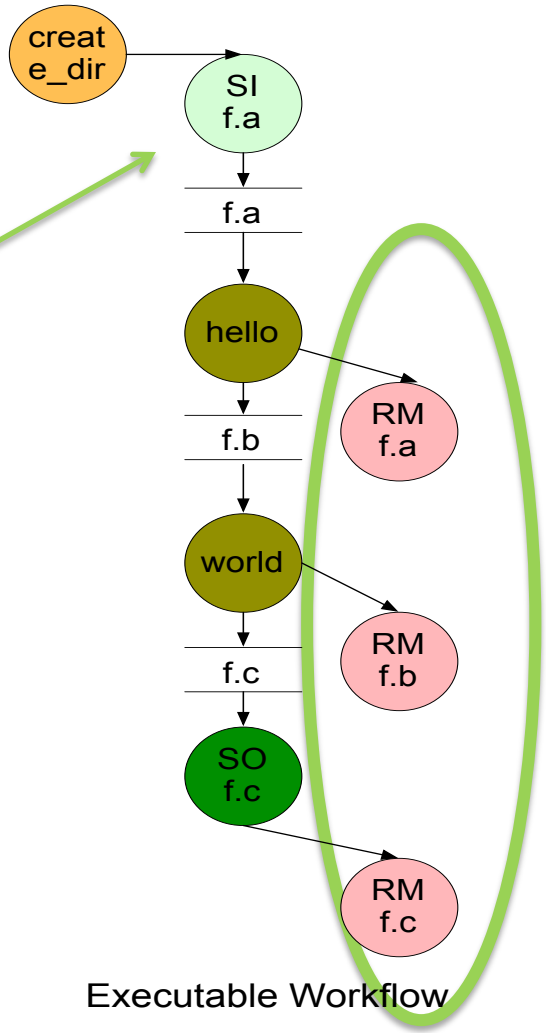


Data Management Issues for Data-Intensive Workflows

- Assumption: Workflow tasks process and generate files
- Input data may be replicated in the environment
- Storage at execution sites may be limited
- Storage at execution sites is heterogeneous
  - Shared filesystems (HPC systems)
  - Non-shared filesystems (HTCondor, Clouds)
- Execution may happen at different computational resources
- Workflow crashes and we don't want to restart the workflow from the beginning
- Workflow changes, but some intermediate data products are still valid
- Need to save results (maybe intermediate data too)
- Need to know how data was created

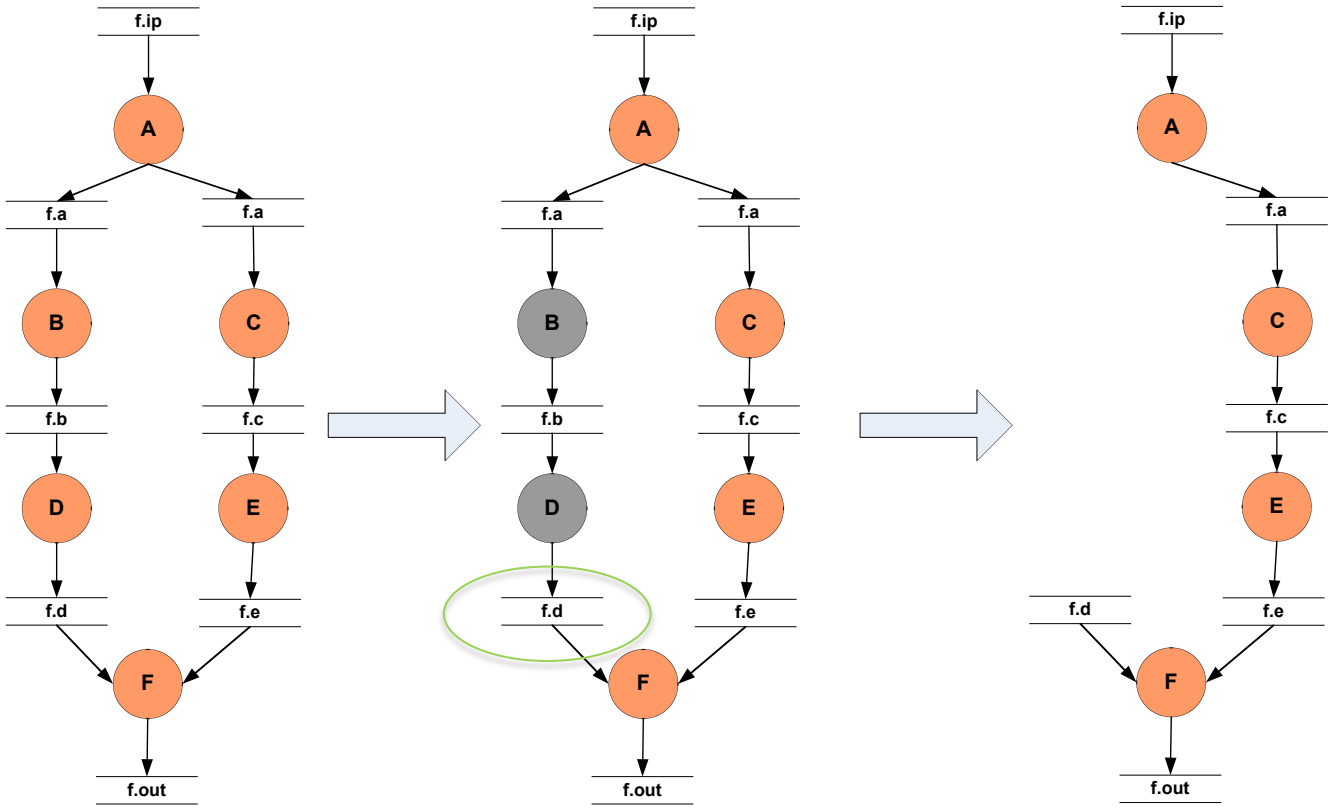
Data Management in Pegasus

- **Input data may be replicated in the environment**
  - **Replica selection during the planning stage**
  - **Can support community preferences via custom replica selection**
- **Storage at execution sites can be limited**



Automatically add tasks to "clean up" data no longer needed

Workflow changes, but some intermediate data products are still valid



Abstract Workflow

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

Delete Job D and Job B

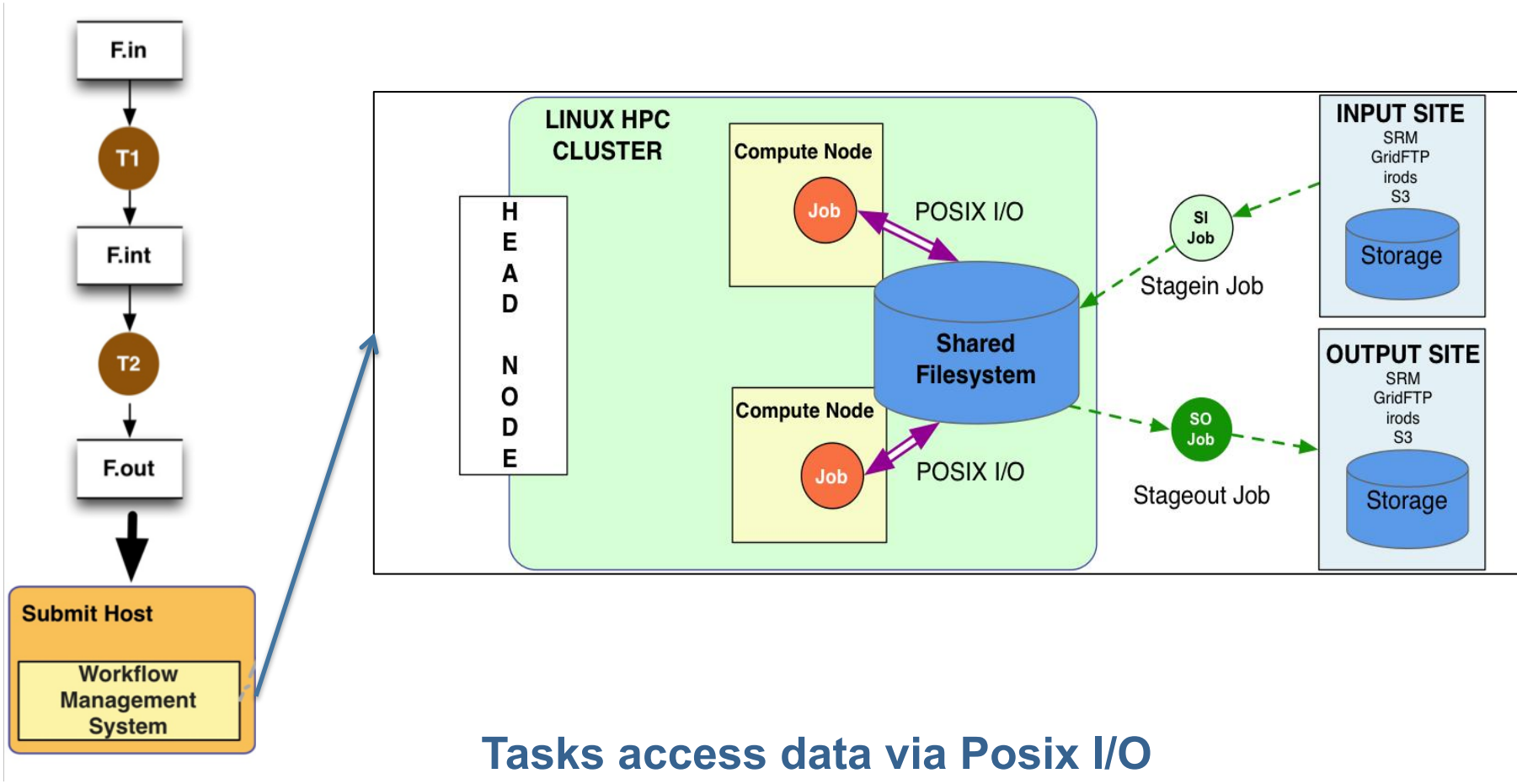
Workflow Reduction

Data Reuse

Workflow-level checkpointing

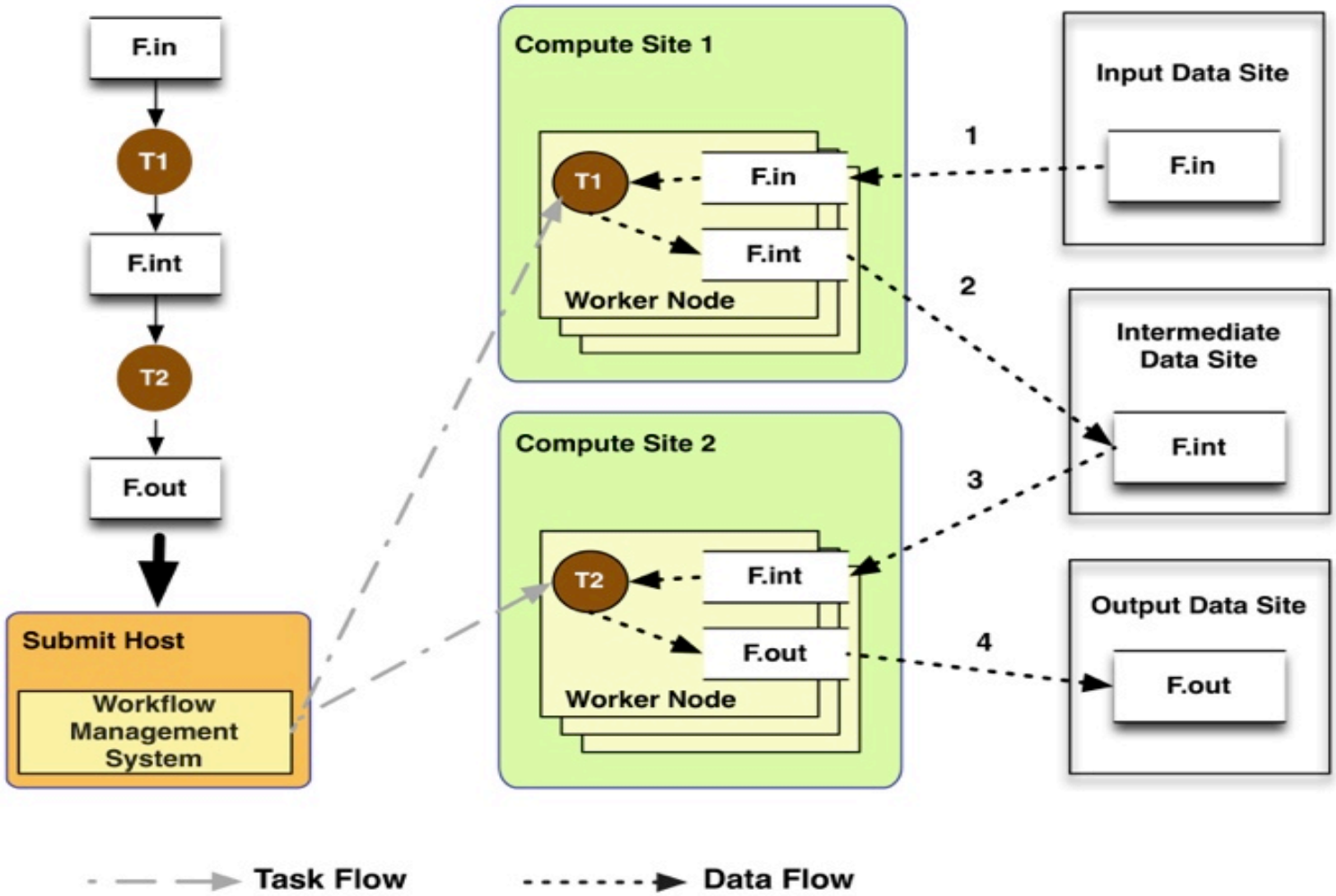
Workflow crashes and we don't want to restart the workflow from the beginning

Storage at execution sites is heterogeneous Shared filesystems



Tasks access data via Posix I/O

Storage at execution sites is heterogeneous
Non-shared filesystems



Supports multi-cloud, cloud and campus, etc. deployments

Also helps with storage limitations

pegasus-transfer subsystem for various storage systems

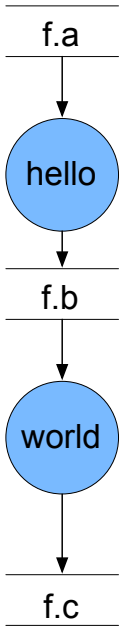
- Command line tool used internally by Pegasus workflows
- Input is a list of source and destination URLs
- Transfers the data by calling out to tools – provided by the system (cp, wget, …) Pegasus (pegasus-gridftp, pegasus-s3) or third party (gsutil)
- Transfers are parallelized
- Transfers between 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

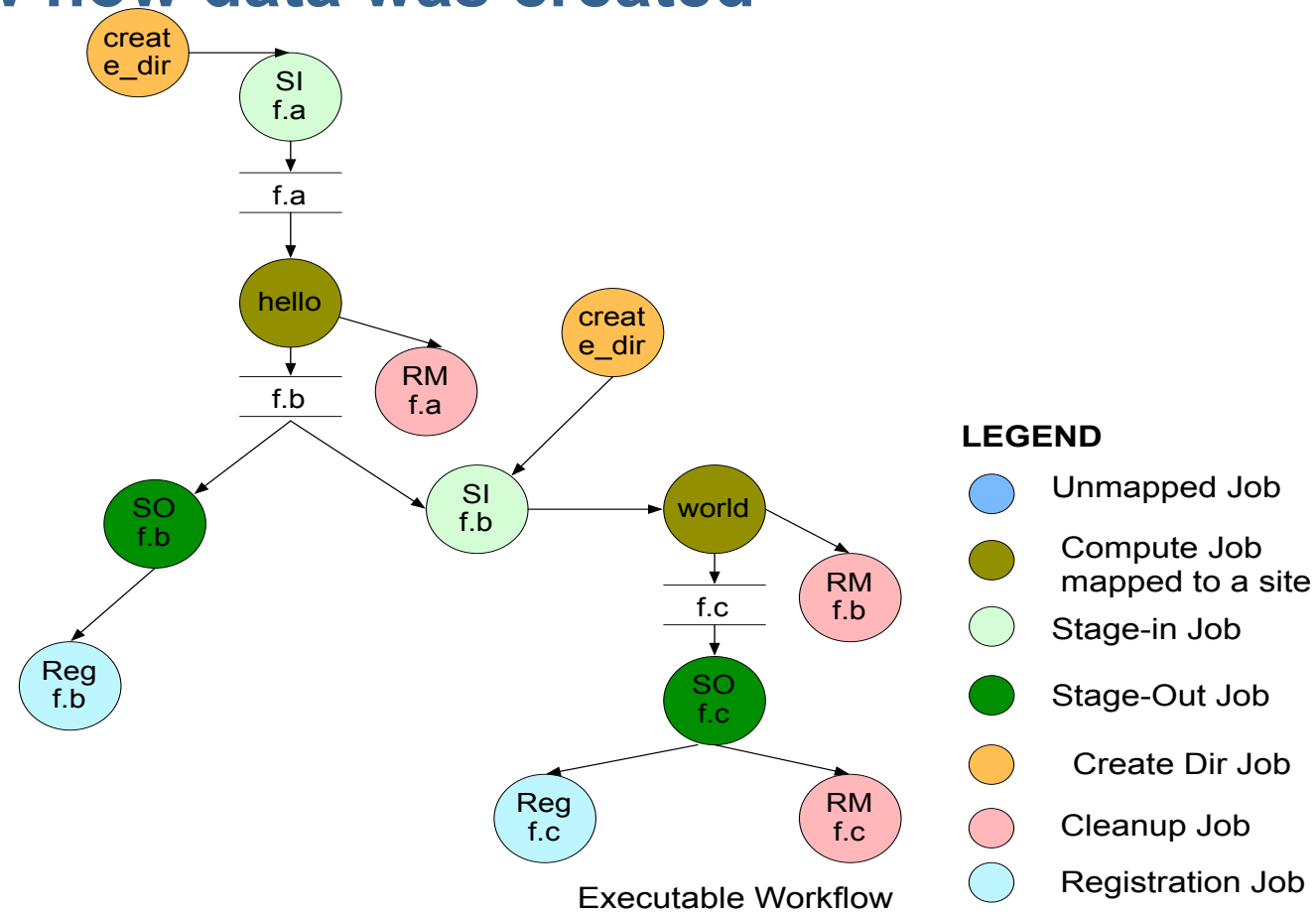
Supported protocols

GridFTP
SRM
iRods
S3
GS
SCP
HTTP
File
Symlink

Execution happens at different resources

Need to save results (maybe intermediate data too)
Need to know how data was created



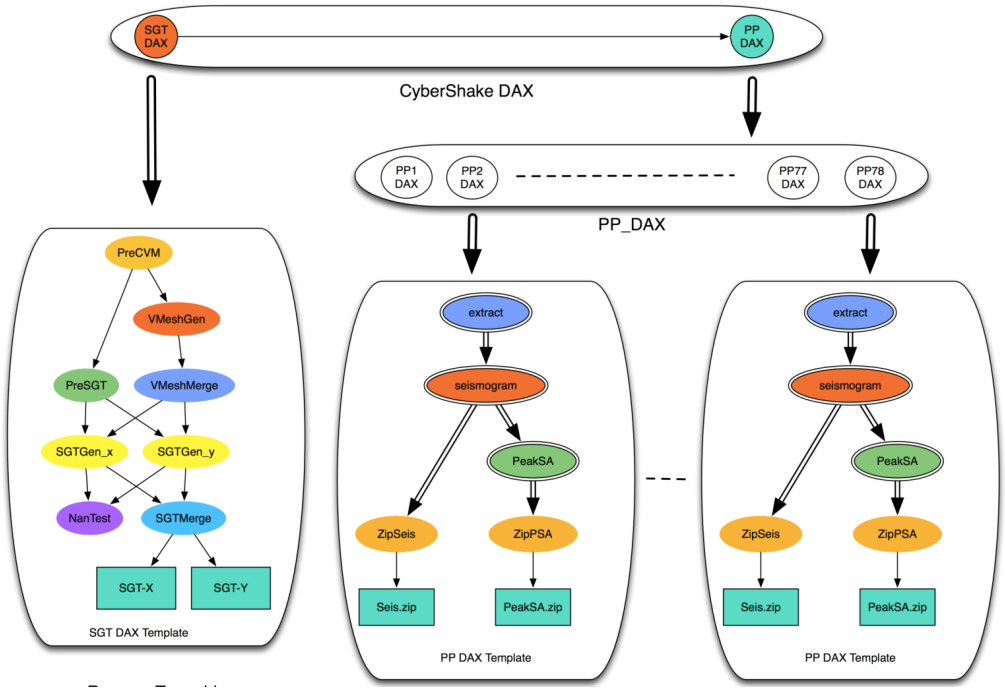


Executable Workflow

Provenance and execution traces are collected and stored in a DB

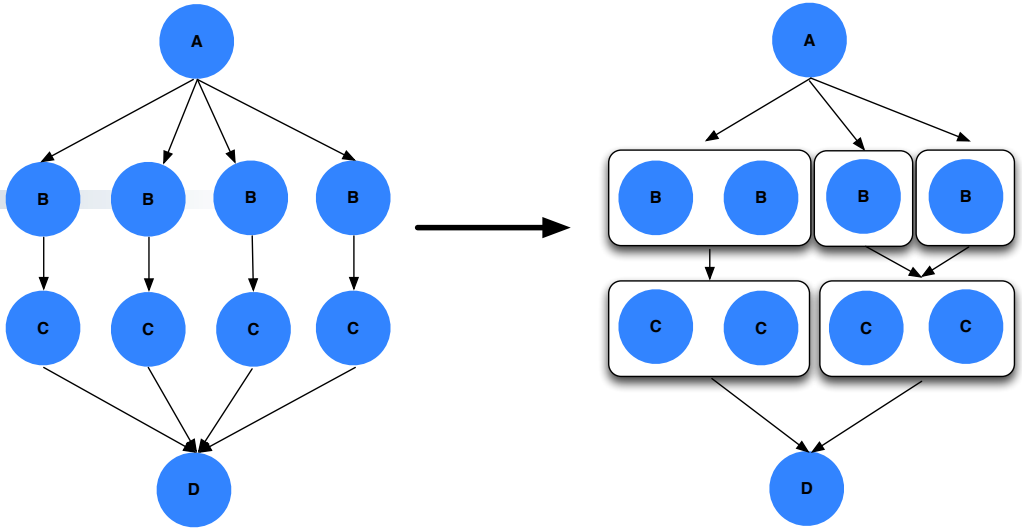
How to Manage Compute Jobs?

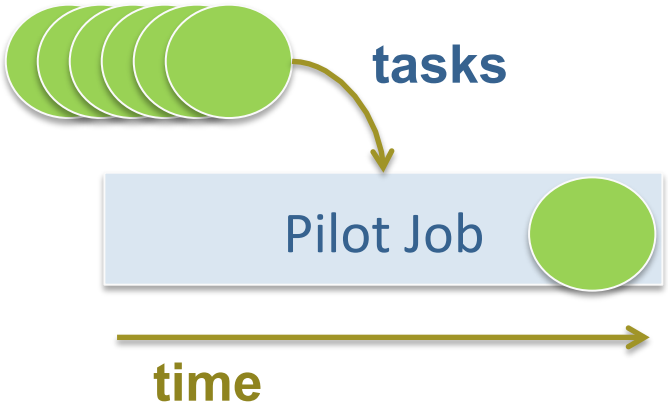
- Mixed workloads, 10s and 100s of TB of data flow
- SCEC's large-scale MPI simulations followed by O(100K) single core post processing jobs



Solutions

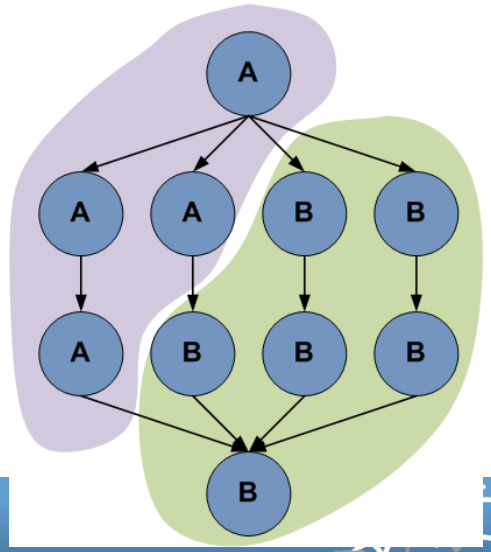
1. Cluster tasks





2. Use "pilot" jobs to dynamically provision a number of resources at a time

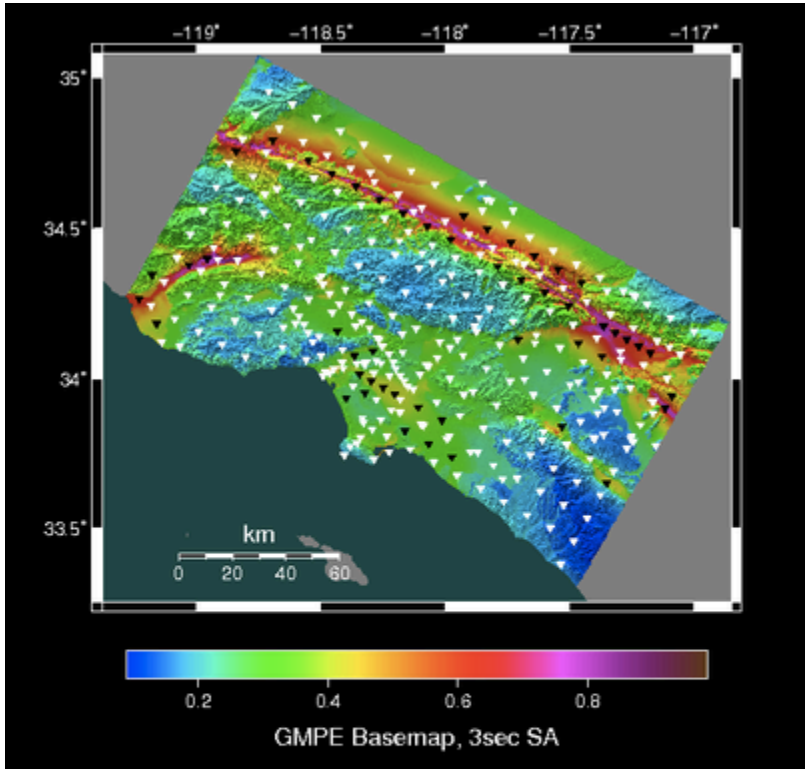
3. Partition the workflow into subworkflows and send them for execution to the target system, use the specialize MPI engine Pegasus MPI_Cluster (PMC)



CyberShake Study 15.4, 2015

Performed CyberShake seismic hazard calculations at 1 Hz seismic frequency, 336 sites

- Used Titan and Blue Waters
- Ran for 38 days
- **Managed 550 TB of data**
  - **197 TB Titan -> Blue Waters**
  - **9.8 TB staged back to USC (~7M files)**
- On average, 53 workflows running concurrently



Blue Waters
Wallclock time: 914.2 hours.
205 sites

Titan
Wallclock time: 896 hrs, 134 sites

Nodes: average 766 (12256 cores), max 14874 (237984 cores, 79.6% of Titan)

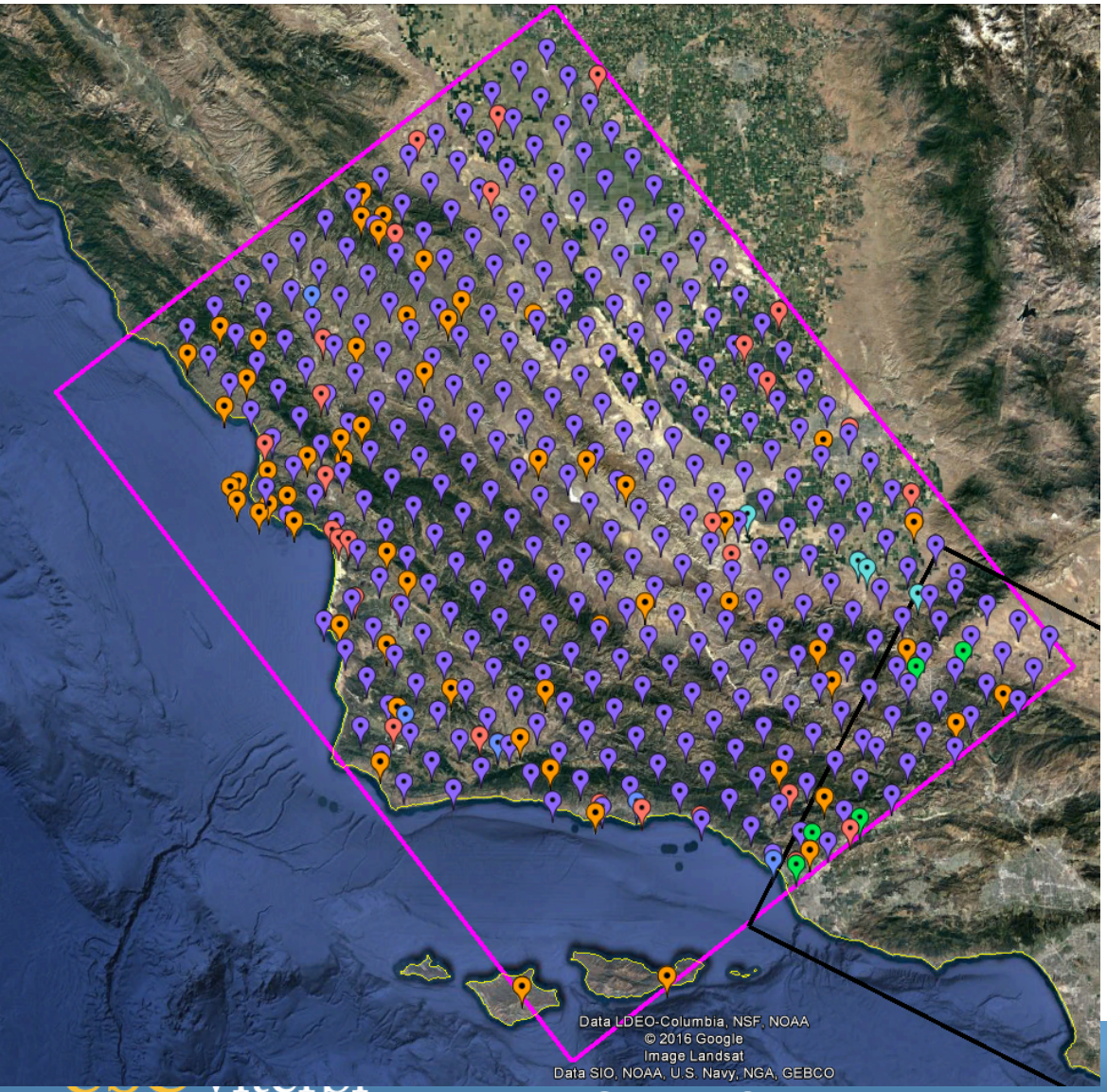
Options for direct HPC job submissions

- Direct job submission from Submit Host/Condor to HPC
  - Requires the workflow management system to run on a head node.
  - Does not work well for cross-site executions
- GRAM job submission to system.
  - Need to run the GRAM gatekeeper on head node
  - Need to open up ports to allow X.509 authenticated clients to submit jobs. Difficult with some system policies. (NERSC uses that)
- HTCondor-CE.
  - Requires HPC system to run Condor on a head node
  - Uses HTCondor-CE for remote job submission
  - Replacement for GRAM on OSG
  - Same open port firewall issues as #2. (prototyped at ALCF and Blue Waters)

Other Options for HPC job submissions

- Pilot jobs
  - User (manually, or using CRON) submits pilot jobs using whatever means you can
  - Pilot jobs start up Condor daemons on the service nodes, which advertise to the Condor Master on the submit host
  - Jobs are wrapped with aprun commands which launch the executables on the right compute nodes
  - Tricky items in the configuration
  - Worked for the last CyberShake production runs, but does not work well, difficult to figure out good pilot job characteristics
- Remote submission using the Reverse GAHP
  - Involves a service running on a head node, that makes an SSH connection back to the submit host to listen for job submissions from the Condor-G daemon on the submit host.
  - A bit hacky, but it works for SCEC on Titan, will be used for next run

Proposed Study CyberShake sites (438) for Fall 2016



- 10 km spacing (purple)
- 5 km spacing (green)
- Pink box is 180x240 km
- Larger area
- New velocity models

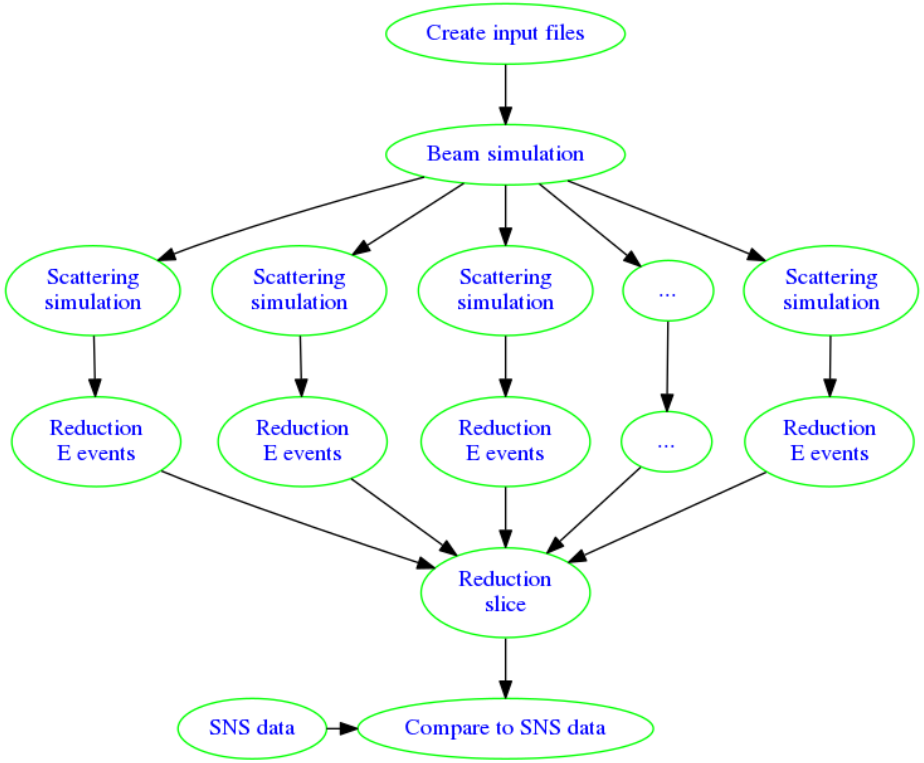
Targeting Titan and Blue Waters

Predict workflow resource needs and workflow performance behavior (DOE Panorama Project)

Aspen performance models (ORNL) supported by simulation (RPI), Correlated monitoring of workflows and infrastructure (LBNL, USC), Anomaly detection (RENCI)

- Compares virtual neutron experiments with real neutron experimental data
- Data from Spallation Neutron Source (SNS)
- Uses McVine code
- Does scattering simulations in parallel for each angle
- Remote simulations done on Edison at NERSC

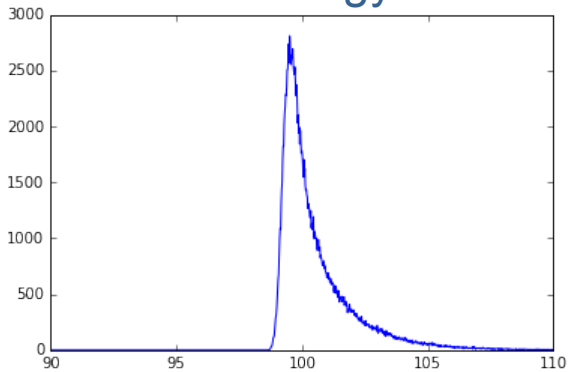
New SNS Panorama Workflow



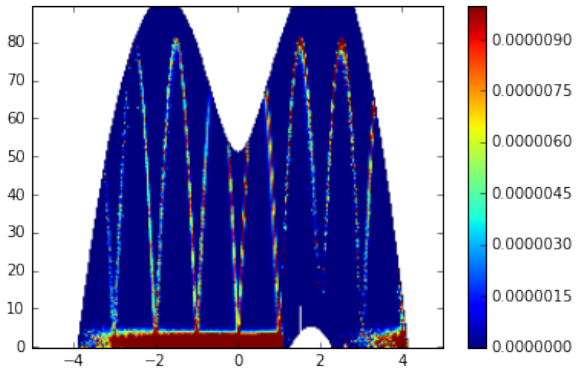
Application: Single Crystal Spin Waves

- **Spin waves are propagating disturbances in the ordering of magnetic materials**

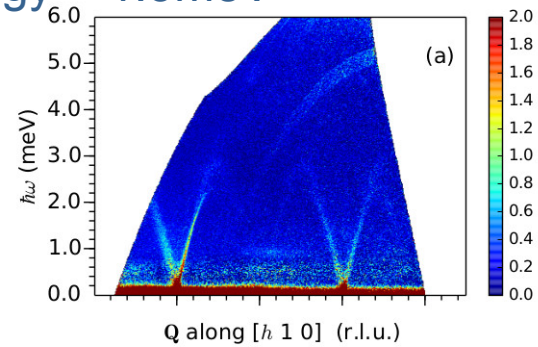
Workflow Test: Beam Energy Profile



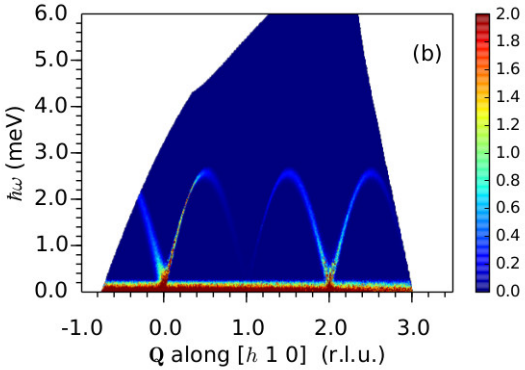
Spin Waves from Workflow

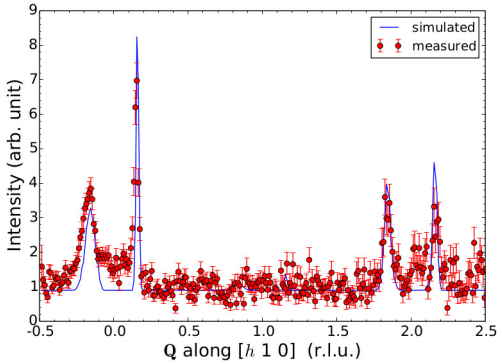


$K_2V_3O_8$: HYSPEC Experiment 1.2meV < Energy < 1.3meV



Virtual Experiment



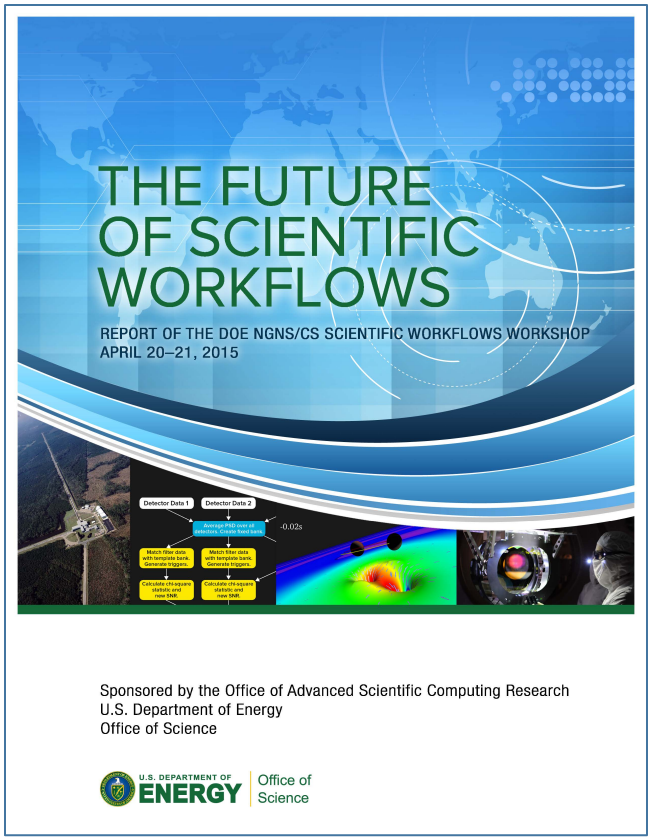


Selected Research Directions

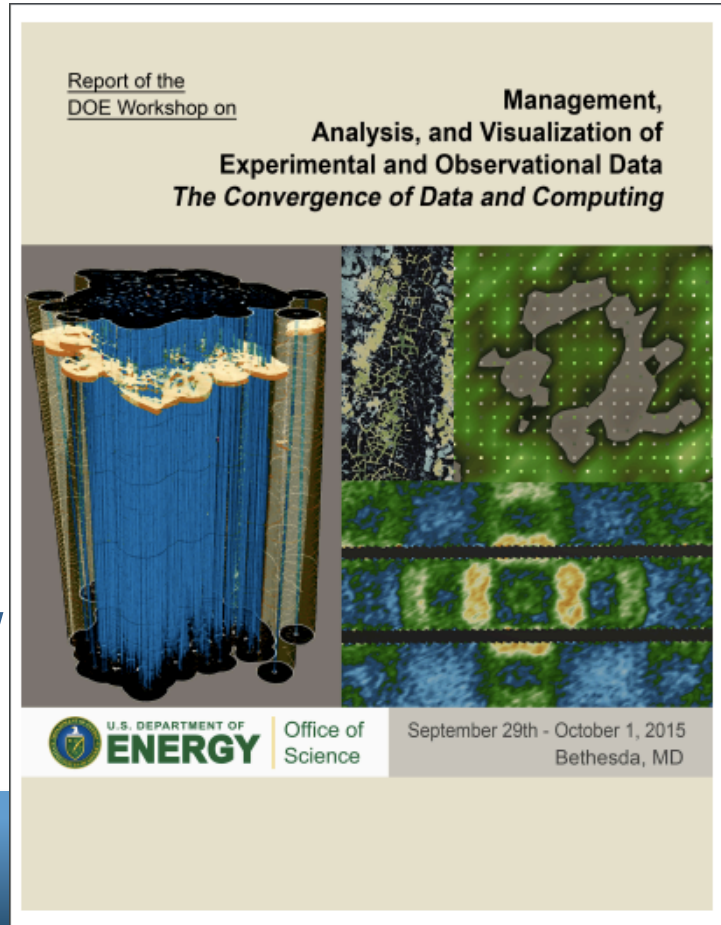
PANORAMA

- Automate resource/performance prediction, resource provisioning, and anomaly detection
- Better integration of resource provisioning/management and workflow management
  - Promote data locality within the HPC system across a workflow
  - Potentially across workflow ensembles
- Enhance MPI-based engine, explore in-situ WMS
  - Take advantage of architectural features (burst buffers)
- Reproducibility, Transparency, Reuse
  - Explore how we can capture, quantify, publish workflows in a reproducible way

Looking into the Future of Scientific Workflows



http://science.energy.gov/~/media/ascr/pdf/programdocuments/docs/workflows_final_report.pdf



http://science.energy.gov/~/media/ascr/pdf/programdocuments/docs/ascr-eod-workshop-2015-report_160524.pdf