What is missing in workflow technologies?

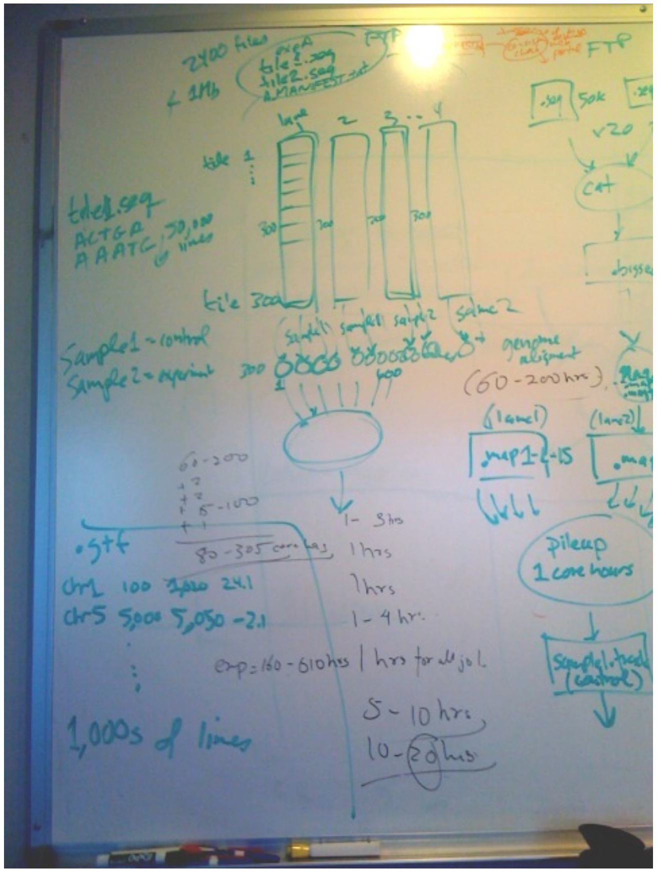
Ewa Deelman, Ph.D

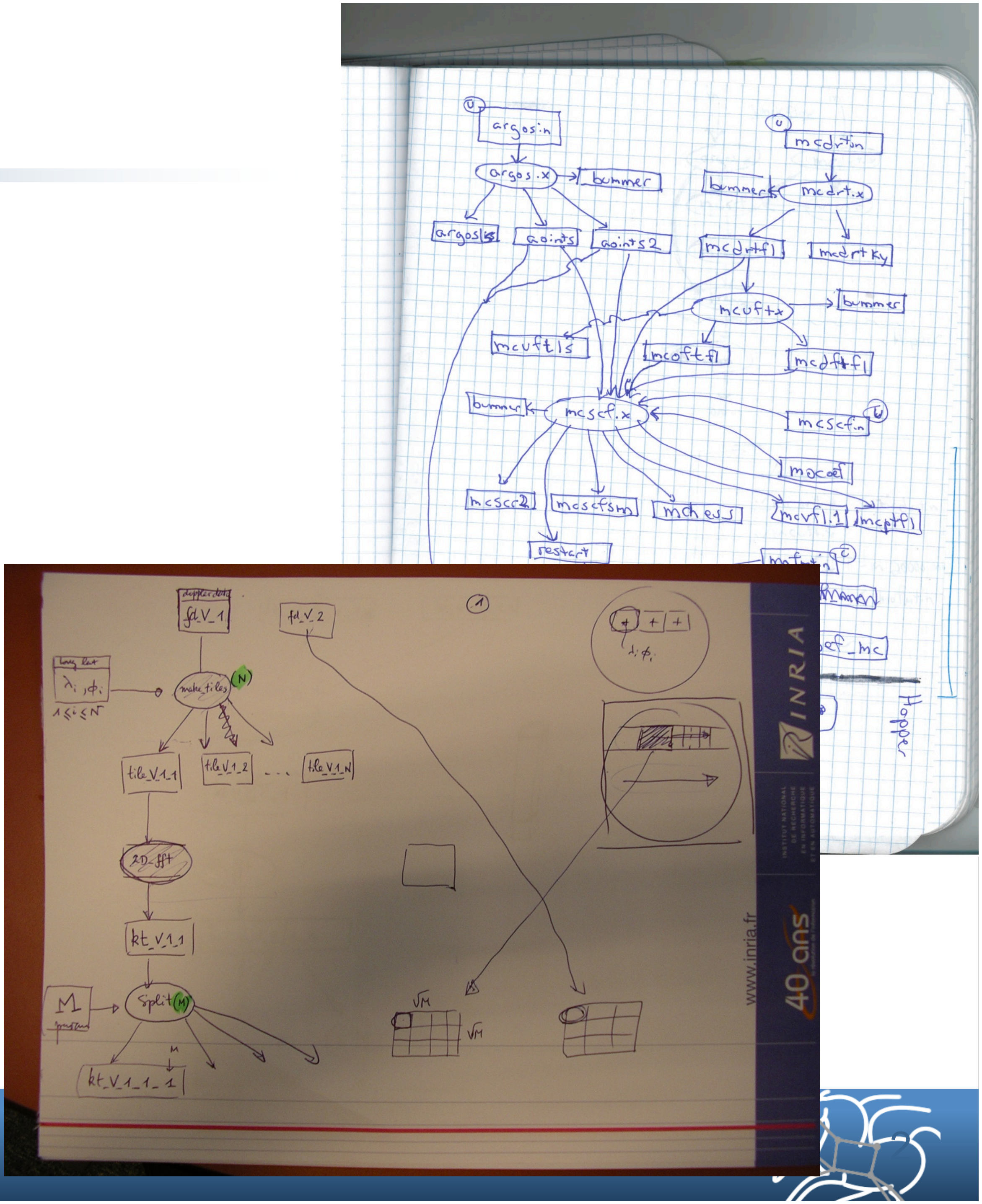
Science Automation Technologies Group

USC Information Sciences Institute

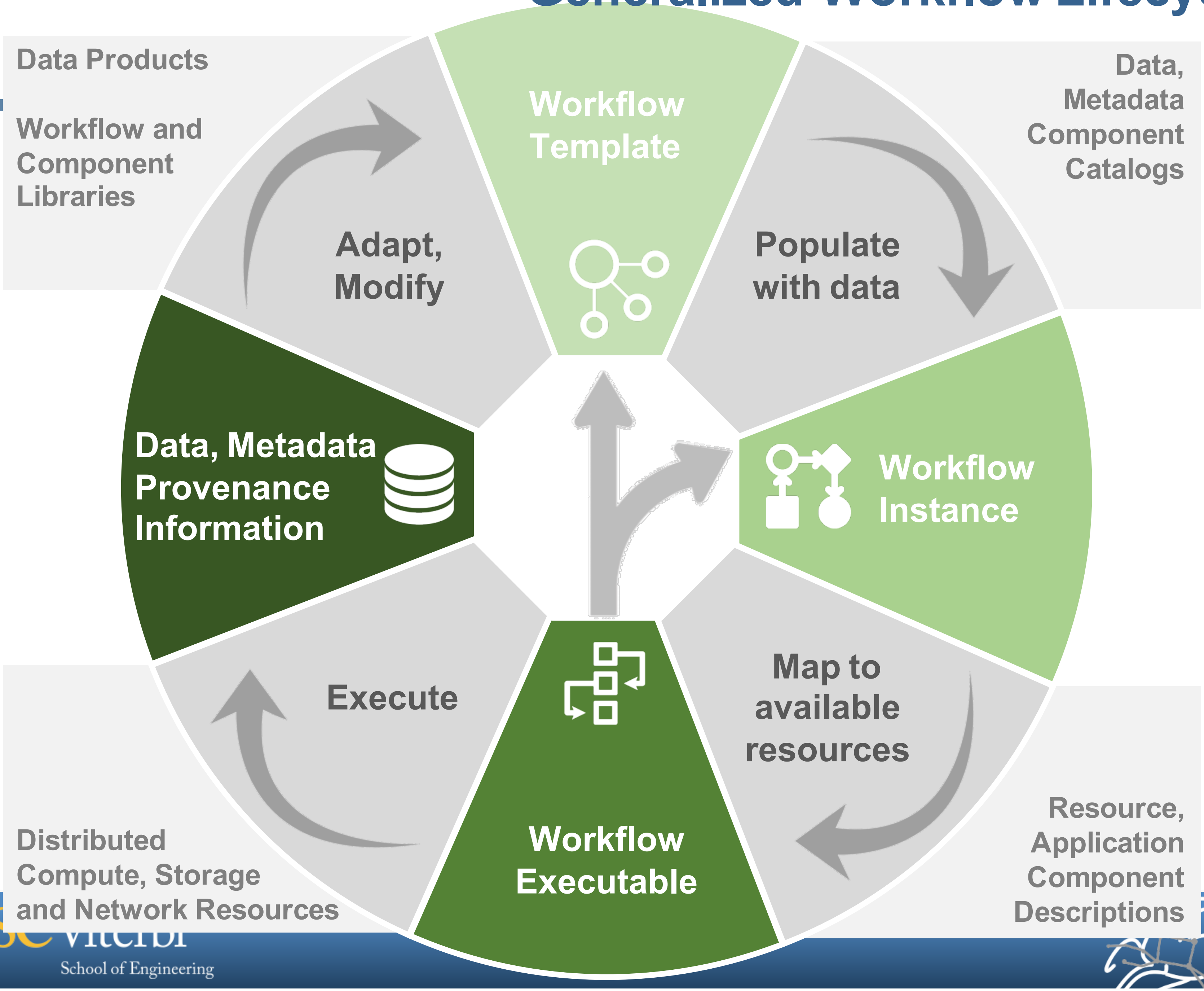
Funding from DOE, NSF, and NIH

The challenge

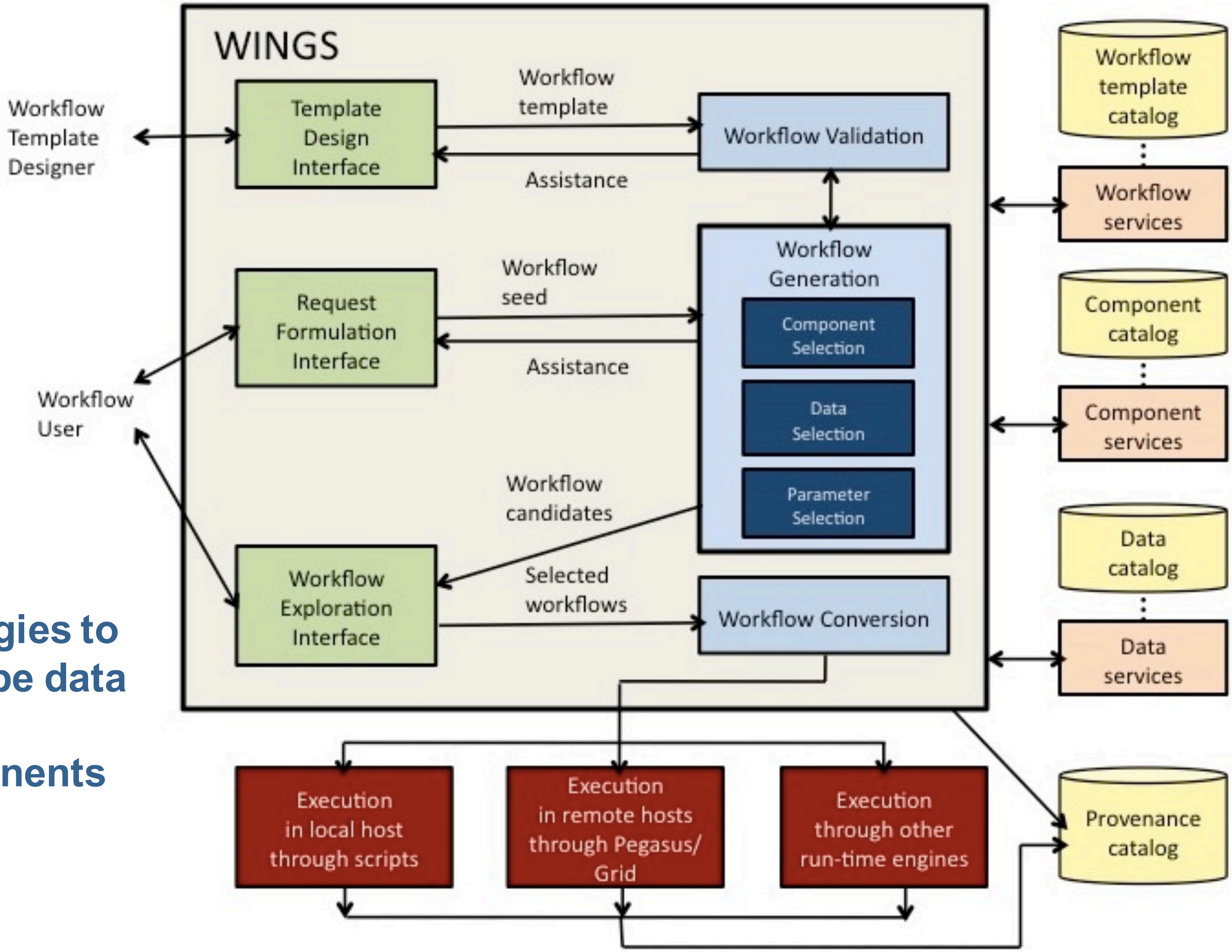




Generalized Workflow Lifecycle


Data Products
Workflow and Component Libraries
Workflow Template
Data, Metadata Component Catalogs
Adapt, Modify
Populate with data
Data, Metadata Provenance Information
Workflow Instance
Execute
Map to available resources
Workflow Executable
Distributed Compute, Storage and Network Resources
Resource, Application Component Descriptions

Workflow Templates

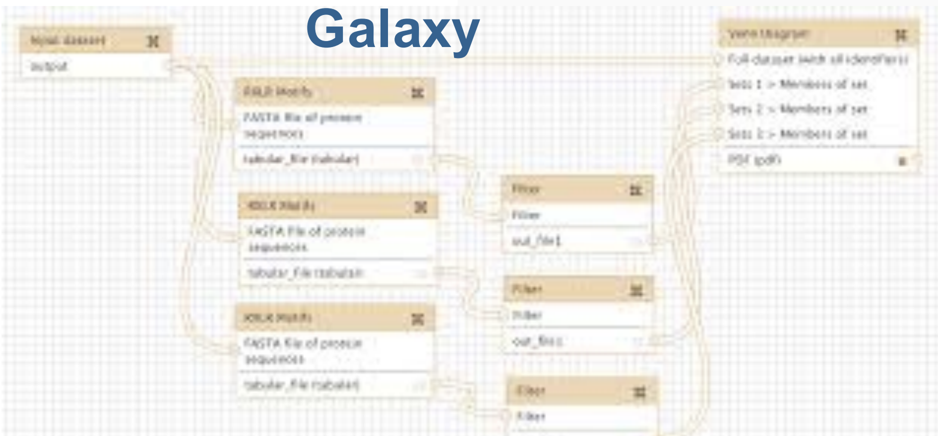


Uses ontologies to describe data and components

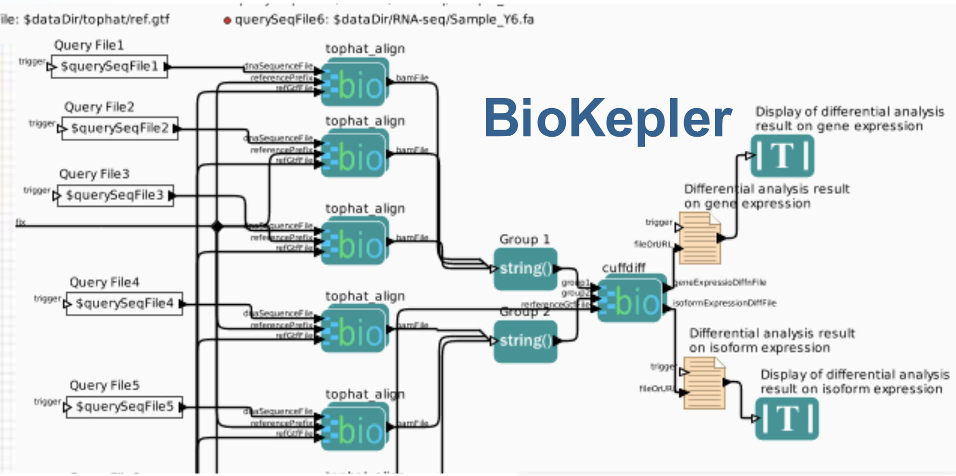
USC

A number of workflow composition frameworks

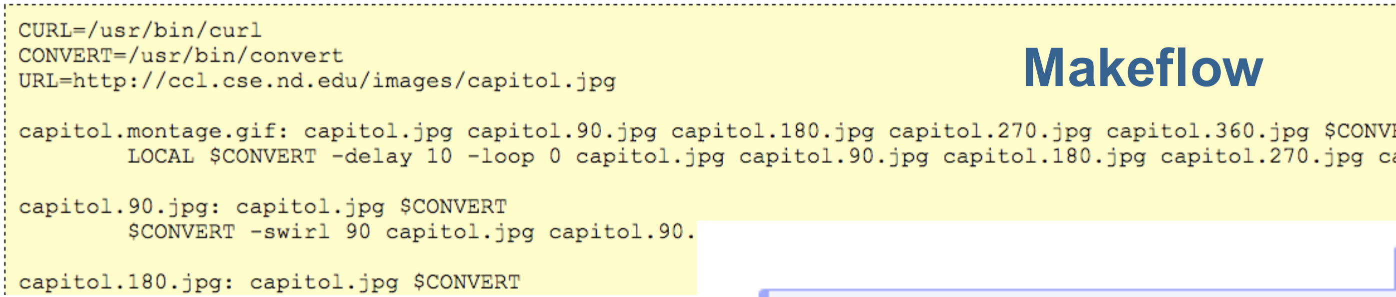
Galaxy



BioKepler


refGtfFile: $dataDir/tophat/ref.gtf
querySeqFile6: $dataDir/RNA-seq/Sample_Y6.fa
Query File1
trigger
$querySeqFile1
Query File2
$querySeqFile2
Query File3
$querySeqFile3
Query File4
$querySeqFile4
Query File5
$querySeqFile5
fix
tophat_align
dnaSequenceFile
referencePrefix
refGtfFile
bamFile
bio
Group 1
string()
Group 2
string()
cuffdiff
group1
group2
referenceGtfFile
geneExpressioDiffnFile
isoformExpressionDiffFile
Differential analysis result on gene expression
Display of differential analysis result on gene expression
Differential analysis result on isoform expression
Display of differential analysis result on isoform expression
fileOrURL
T

Makeflow

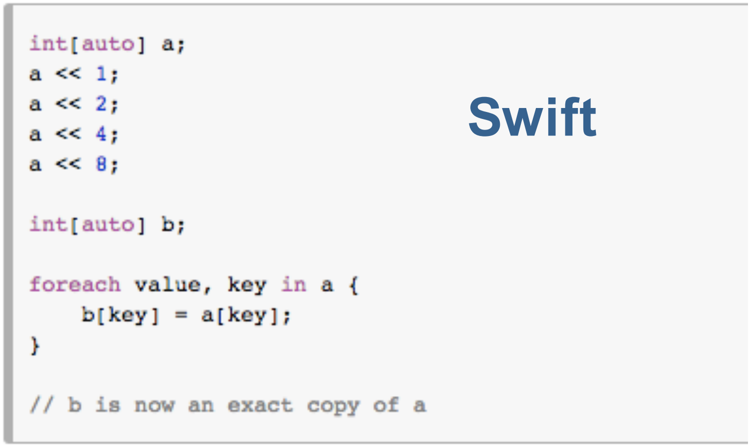


```
CURL=/usr/bin/curl
CONVERT=/usr/bin/convert
URL=http://ccl.cse.nd.edu/images/capitol.jpg

capitol.montage.gif: capitol.jpg capitol.90.jpg capitol.180.jpg capitol.270.jpg capitol.360.jpg $CONVERT
        LOCAL $CONVERT -delay 10 -loop 0 capitol.jpg capitol.90.jpg capitol.180.jpg capitol.270.jpg capito

capitol.90.jpg: capitol.jpg $CONVERT
        $CONVERT -swirl 90 capitol.jpg capitol.90.

capitol.180.jpg: capitol.jpg $CONVERT
```

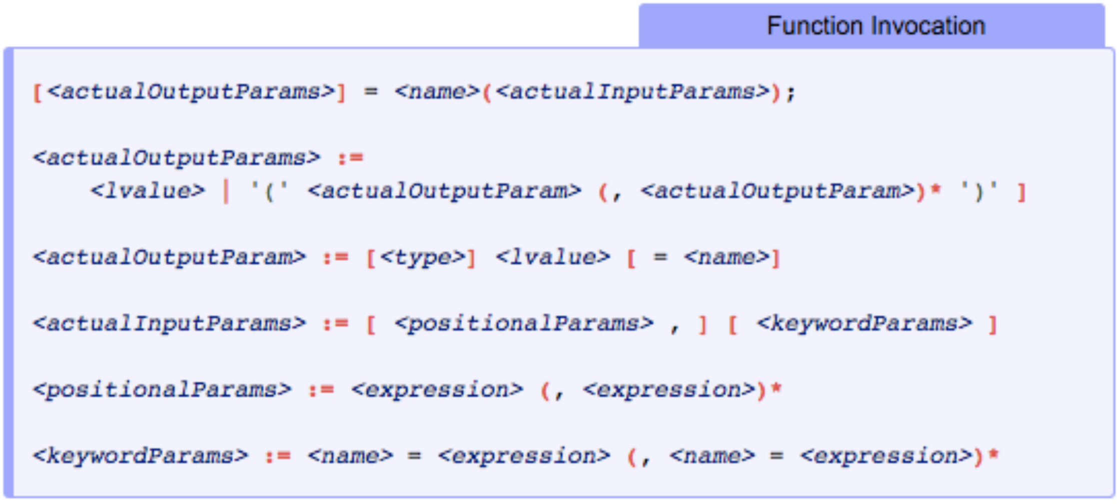
Java,
Perl,
Python,
R

Pegasus

Swift



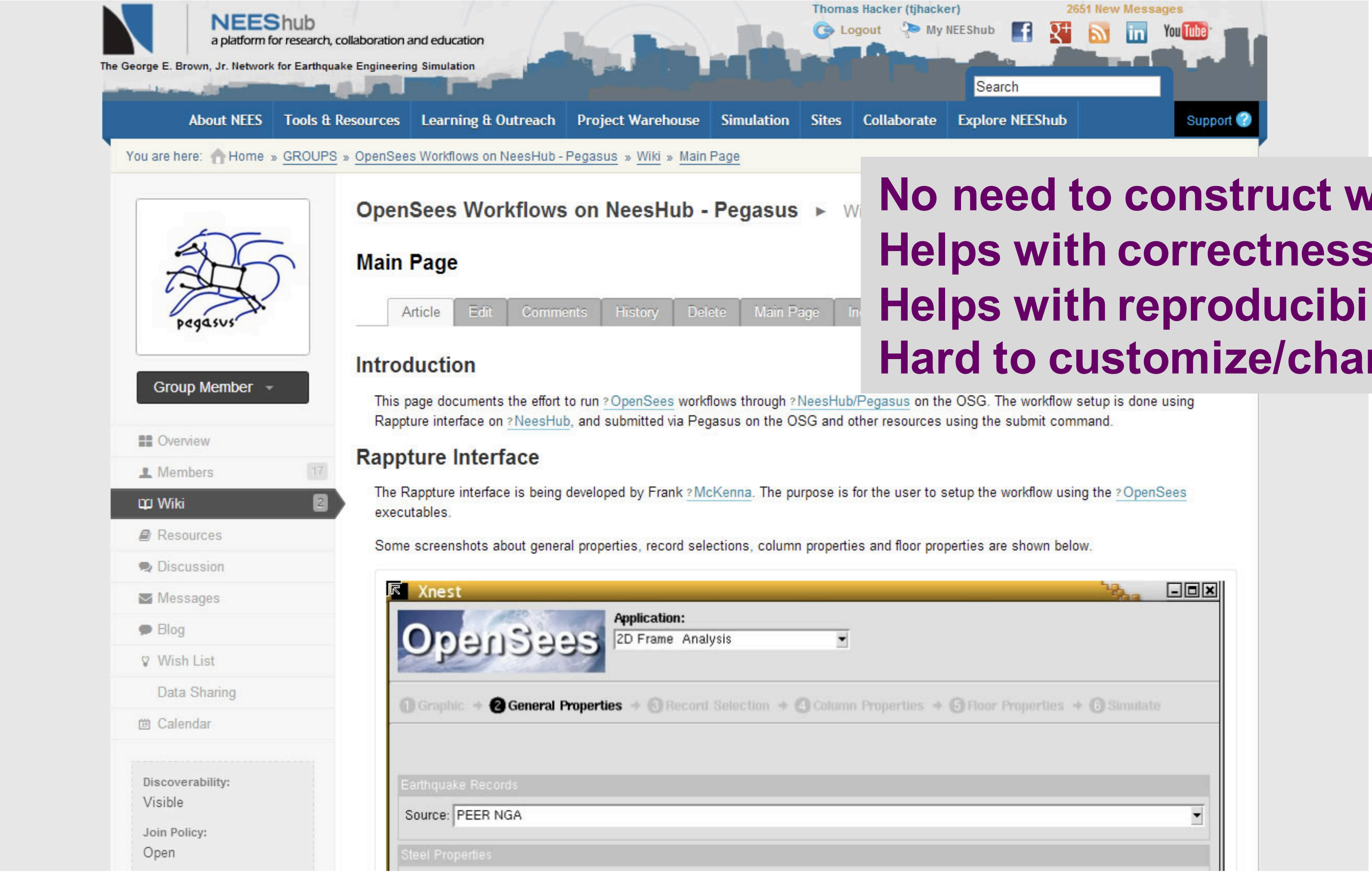
```
int[auto] a;
a << 1;
a << 2;
a << 4;
a << 8;

int[auto] b;

foreach value, key in a {
    b[key] = a[key];
}

// b is now an exact copy of a
```



Function Invocation

```
[<actualOutputParams>] = <name>(<actualInputParams>);

<actualOutputParams> :=
    <lvalue> | '(' <actualOutputParam> (, <actualOutputParam>)* ')' ]

<actualOutputParam> := [<type>] <lvalue> [ = <name>]

<actualInputParams> := [ <positionalParams> , ] [ <keywordParams> ]

<positionalParams> := <expression> (, <expression>)*

<keywordParams> := <name> = <expression> (, <name> = <expression>)*
```

Sometime the workflows is behind a portal



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

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

Discoverability:
Visible

Join Policy:
Open

OpenSees Workflows on NeesHub - Pegasus ▸ Wi

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

Source: PEER NGA

No need to construct workflow
Helps with correctness
Helps with reproducibility
Hard to customize/change

Mapping of of Workflows onto Resources -- Provisioning

- Traditionally resources were already provisioned: XSEDE clusters, DOE LCF systems
  - Need to submit jobs to a queue and make sure the input data is there
  - Lack of storage provisioning capability or provisioning particular nodes
- Opportunistic resources used with on-the-fly provisioning
  - Open Science Grid: HTC Condor Glideins, GlideinWMS
  - When you land on a resource, need to pull data and push out results before
  - Needs robust fault recovery, as resources can disappear
  - Lack of storage provisioning capability, lacks network provisioning
- Cloud-based environments
  - Need to know what to provision and for how long
  - Need to adjust the provisioning over time
  - Need to keep an eye on costs (stay within a budget, minimize the costs)
  - Need a fail-safe mechanism to deprovision resources when no longer needed or when things go wrong

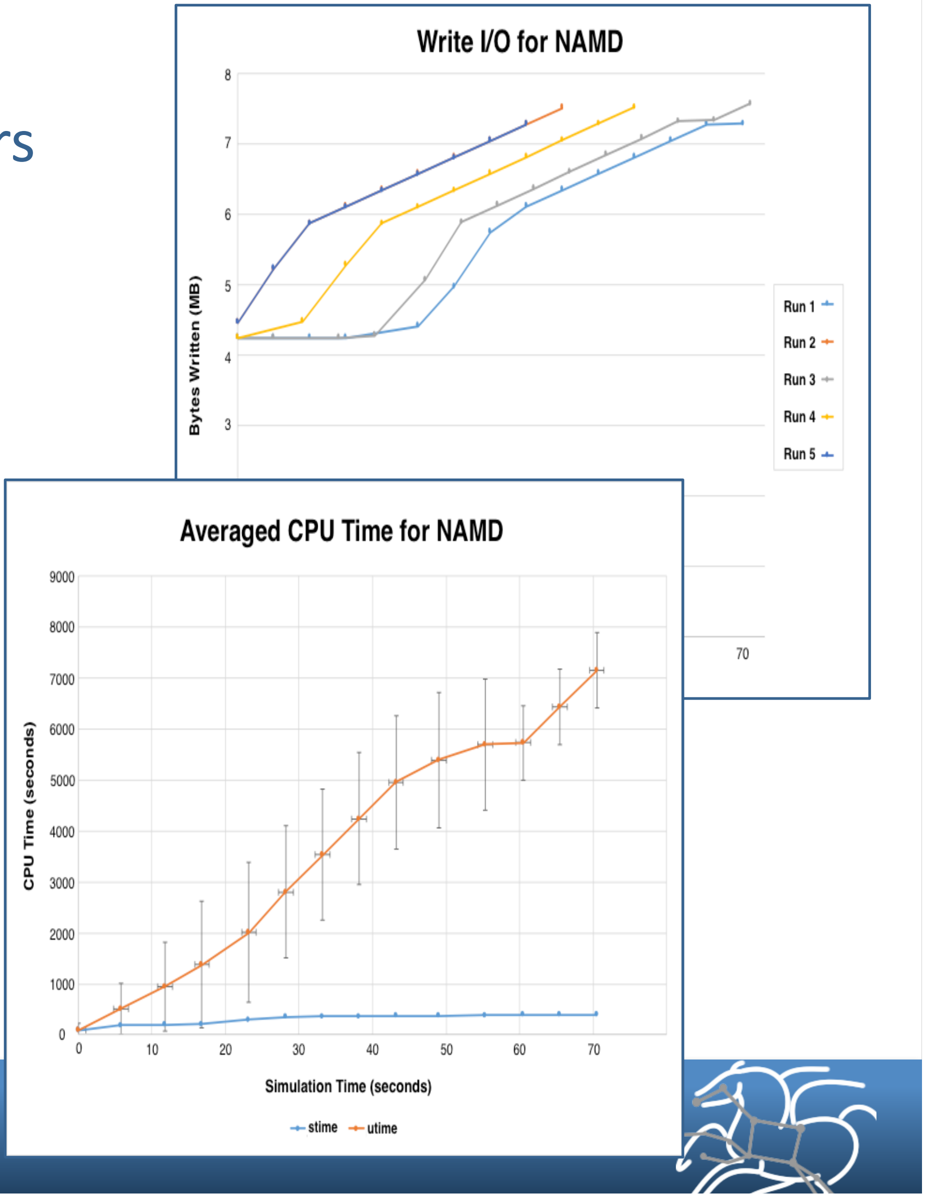
Determining the needed resources

Application Monitoring

- CPU, I/O, memory, perf counters
- Function interposition
- MPI and serial jobs
- Real-time reporting

Infrastructure Monitoring

- Load, disk I/O, network, etc.
- Standard tools
- Data stored in time series DB



Analytical Modeling Example for SNS workflow with ASPEN

MD template model

```
model NAMD_Template {
  // application parameters
  // (defined in the input file)
  param nAtoms      = 1e6
  param nTimeSteps = 100
  // solve for these parameters
  // (within the given ranges)
  param c = 1 in 1 .. 1e18
  param d = 1 in 1 .. 1e18
  // application behavior:
  // execution and control flow
  kernel main
  {
    iterate [nTimeSteps] {
      execute {
        loads [c * nAtoms^2]
        flops [d * nAtoms]
      }
    }
  }
}
```

+

CSV data file with parameters and runtimes

nAtoms	nTimeSteps	nCores	machine	runtime
1e6	100	144	exogeni	384.2
1e6	100	144	hopper	340.1
1e6	150	144	hopper	482.9

=

Concrete NAMD model

```
model NAMD_Equilibrate {

  // NAMD input parameters
  param nAtoms      = 1e6
  param nTimeSteps  = 100

  // calculation-specific constants
  param c = 402.1
  param d = 10.95

  // NAMD application behavior
  kernel main
  {
    iterate [nTimeSteps] {
      execute {
        loads [c * nAtoms^2]
        flops [d * nAtoms]
      }
    }
  }
}
```

nAtoms and nTimeSteps defined in template application model and CSV input data

nCores defined in machine models and CSV input data

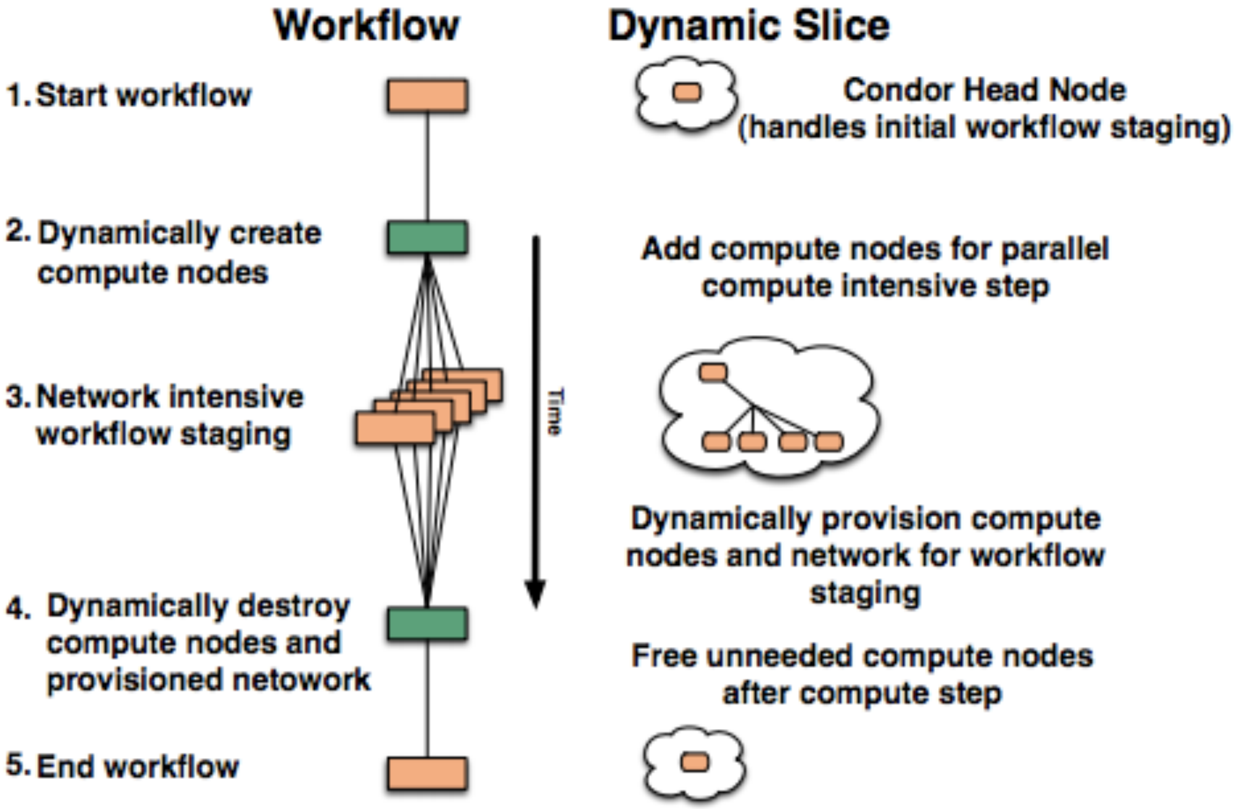
solves for c and d, filling out a concrete application model for that problem

new predictions can still vary nAtoms, nTimeSteps, and nCores

USC Viterbi School of Engineering

Work with Jeff Vetter, ORANL

Interleaving Workflow Management and Provisioning



Workflow

Dynamic Slice

1. Start workflow

Condor Head Node
(handles initial workflow staging)

2. Dynamically create compute nodes

Add compute nodes for parallel compute intensive step

3. Network intensive workflow staging

Dynamically provision compute nodes and network for workflow staging

4. Dynamically destroy compute nodes and provisioned netowork

Free unneeded compute nodes after compute step

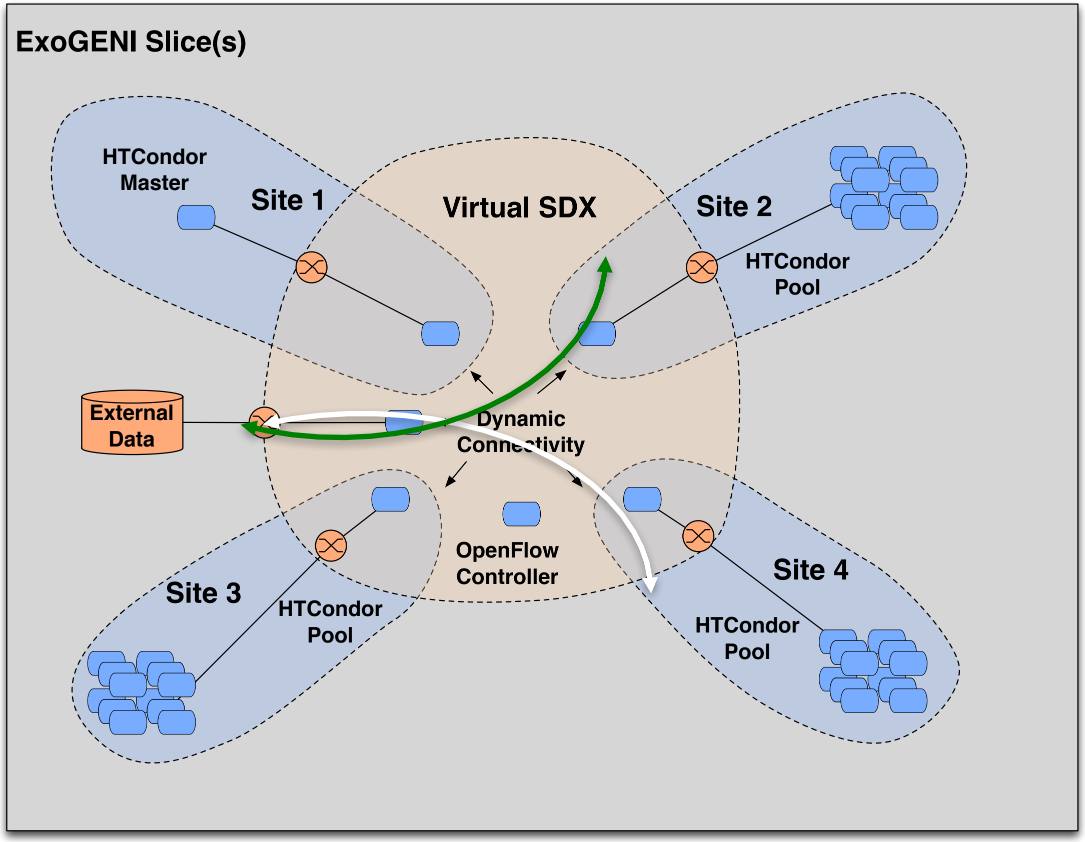
5. End workflow

Work Jeff Chase (ORCA), Anirban Mandal, Paul Ruth, Ilya Baldin

ExoGENI Virtual SDX: Panorama Workflows

PANORAMA

Software Defined Exchanges – meeting point of networks to exchange traffic, securely and with QoS, using SDN protocols



- Panorama modeling and simulation tools enable Pegasus to monitor and manipulate network connectivity & performance
- Virtual SDX **transparently arbitrates prioritized workflow data flows** communicated by Pegasus
- Advanced SDX capabilities can monitor and detect network anomalies, and take adaptation actions.

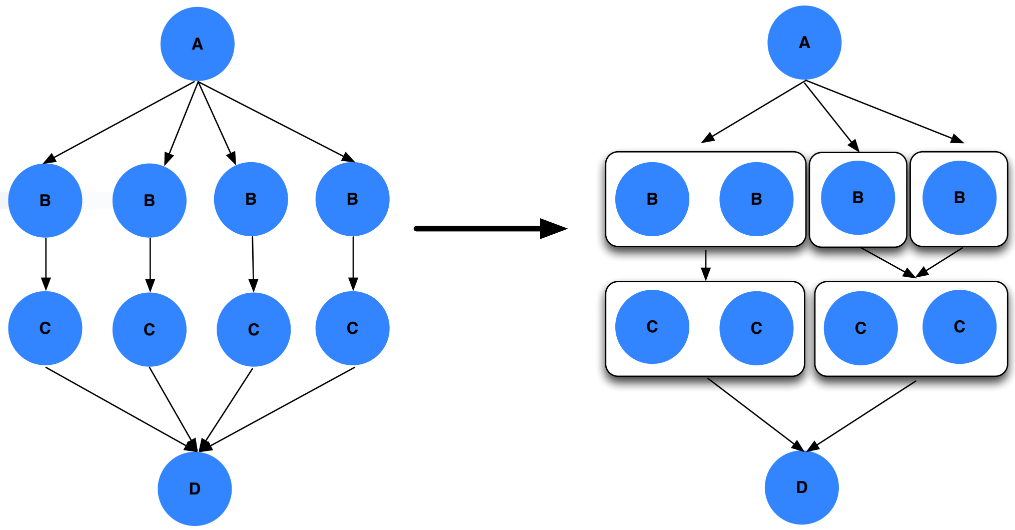
Work Jeff Chase (ORCA), Anirban Mandal, Paul Ruth, Ilya Baldin

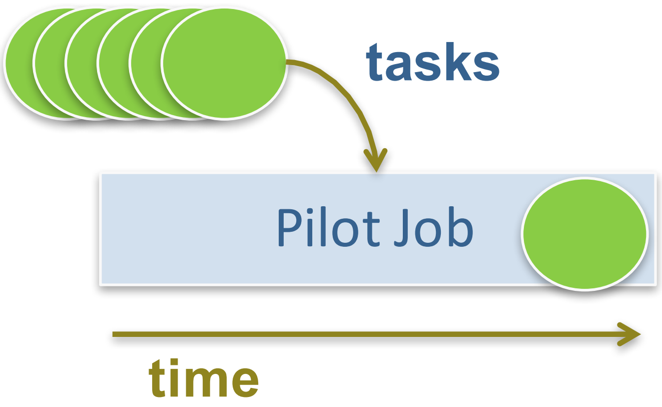
Resource Selection Issues in Distributed Area

- Discover what resources (computation, data, software) are available (or what resources were provisioned)
- Select the appropriate resources based on a architecture, availability of software, performance, reliability, availability of cycles, storage,.. (or provision)
- 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
- Issues of compute "close" to the data, traditionally data moves
  - Extract subsets, compress, pre-compute some values at the source
- Issues of recompute vs retrieve the results
- Managing data access within a workflow and across workflow ensembles – supporting data reuse

Matching Workload to Target System

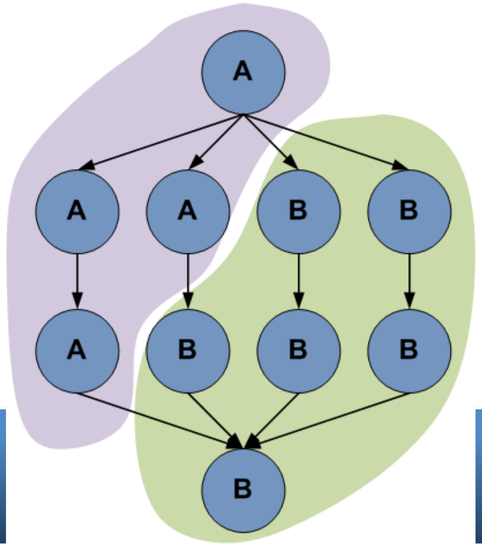
Cluster tasks





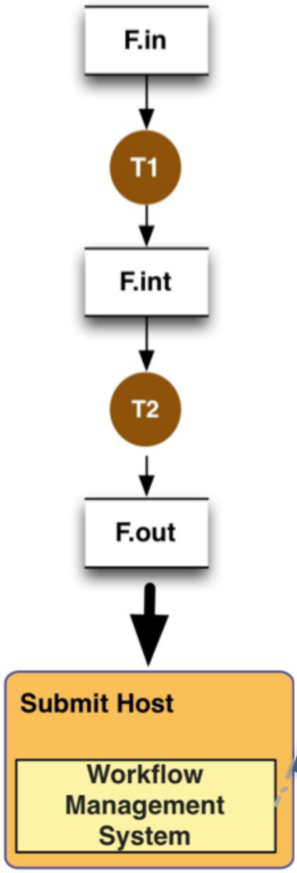
Use "pilot" jobs to dynamically provision a number of resources at a time

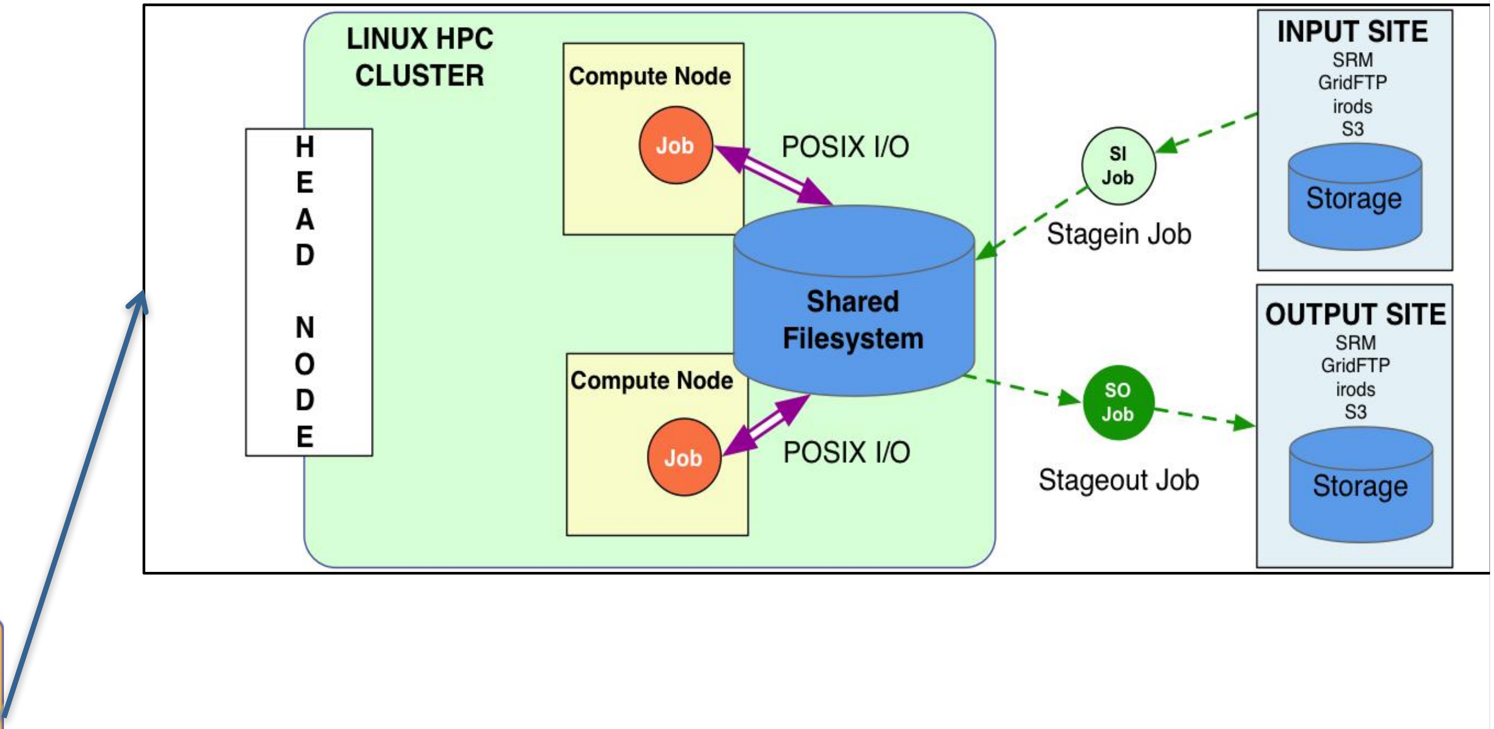
Partition the workflow into subworkflows and send them for execution to the target system



Make Data Flow Over Heterogeneous Fabric

Typical Deployment in Clusters, sharing data through the file system

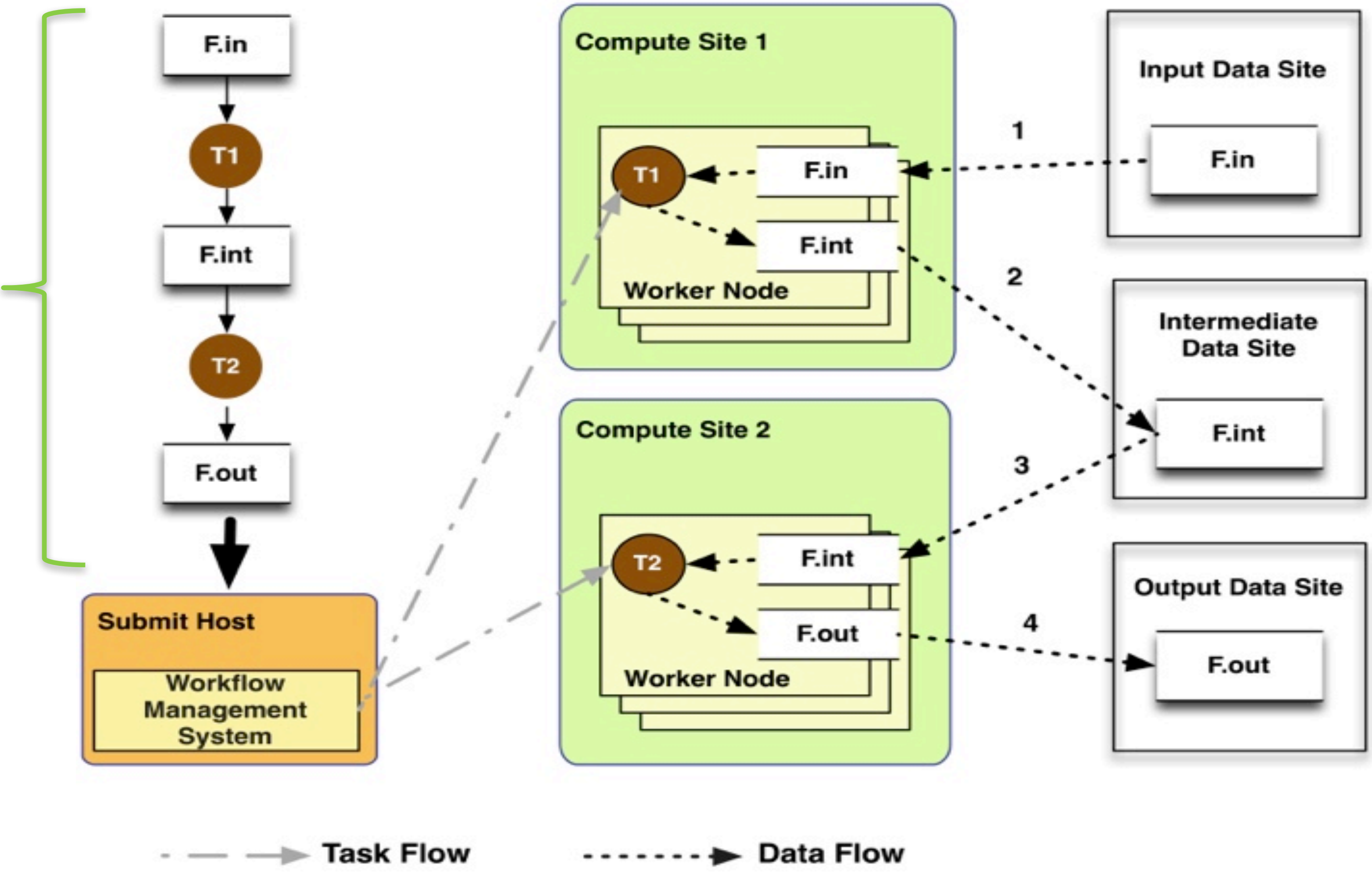




Tasks access data via Posix I/O

Variety of file system deployments: shared vs non-shared

User workflow



F.in

T1

F.int

T2

F.out

Submit Host

Workflow Management System

Compute Site 1

T1

F.in

F.int

Worker Node

Compute Site 2

T2

F.int

F.out

Worker Node

Input Data Site

F.in

Intermediate Data Site

F.int

Output Data Site

F.out

1

2

3

4

Task Flow

Data Flow

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

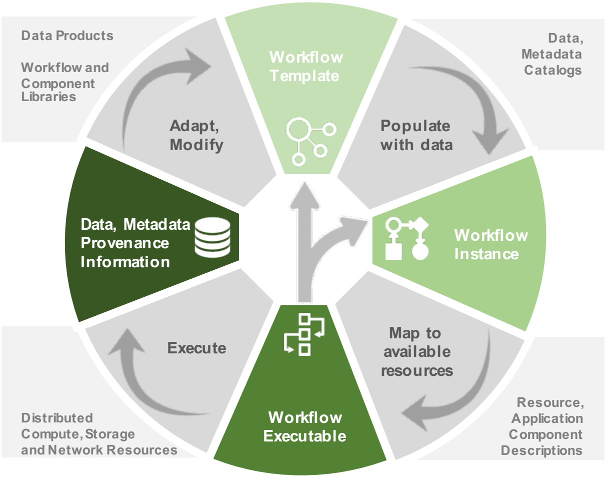
Supported protocols

GridFTP
SRM
iRods
S3
GS
SCP
HTTP
File
Symlink

What’s needed in Mapping and Execution

- Issues of efficiency (time, cost, energy)
- Security of data being managed, using trusted resources (new project with Von Welch and Ilya Baldin)
- Error reporting
  - Easy to interpret, maybe involve machine learning to better categorize errors
- Anomaly detection (Panorama Project)
  - In prototype, but need the technologies to do it in production

Metadata, Provenance information


Data Products
Workflow and Component Libraries
Adapt, Modify
Workflow Template
Populate with data
Data, Metadata Catalogs
Data, Metadata Provenance Information
Workflow Instance
Execute
Workflow Executable
Map to available resources
Distributed Compute, Storage and Network Resources
Resource, Application Component Descriptions

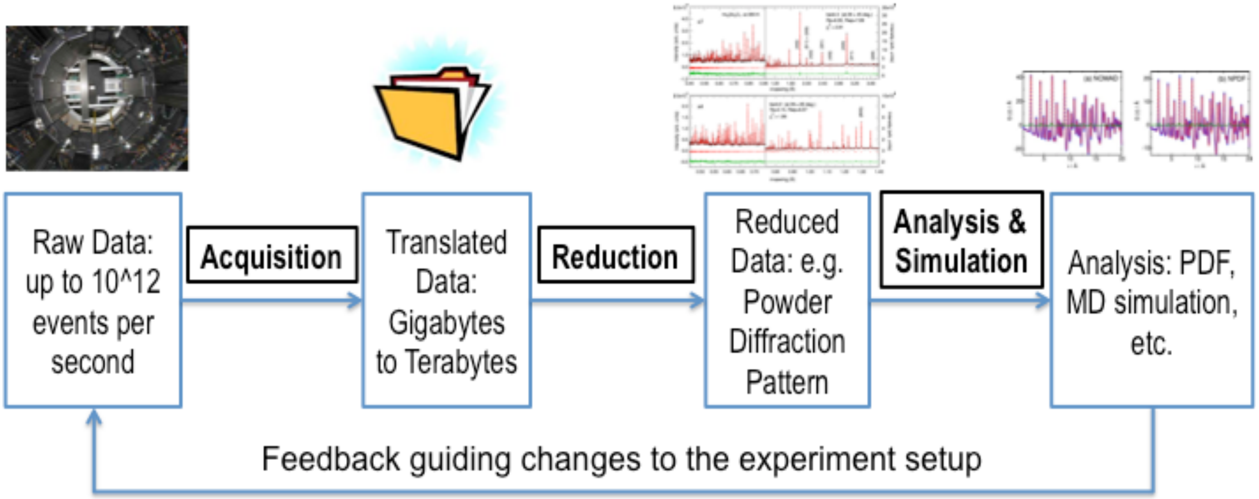
- Some provenance capture standards
- Limited or no tools for provenance exploration
- Limited use of metadata when doing workflow composition

- Now more pressing issues of scientific collaboration
  - Transparency
  - Re-use
  - Reproducibility

Future Applications: Near real time feedback, human-in-the loop

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

More in-situ workflow management



Raw Data: up to 10^12 events per second

Acquisition

Translated Data: Gigabytes to Terabytes

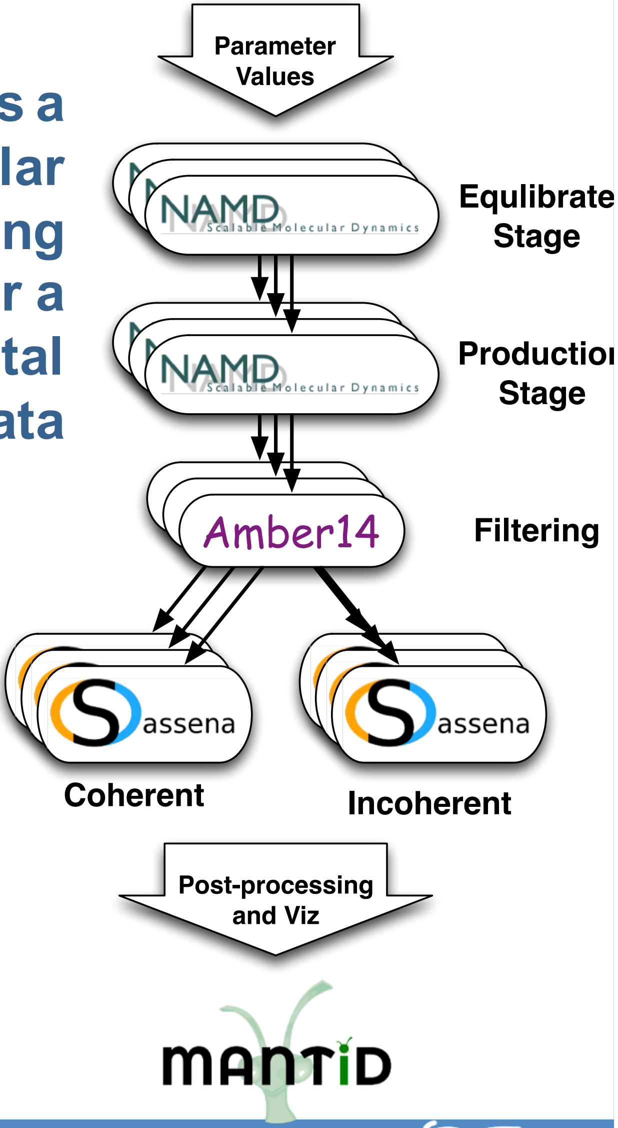
Reduction

Reduced Data: e.g. Powder Diffraction Pattern

Analysis & Simulation

Analysis: PDF, MD simulation, etc.

Feedback guiding changes to the experiment setup



Parameter Values

NAMD Scalable Molecular Dynamics

Equlibrate Stage

NAMD Scalable Molecular Dynamics

Production Stage

Filtering

Coherent

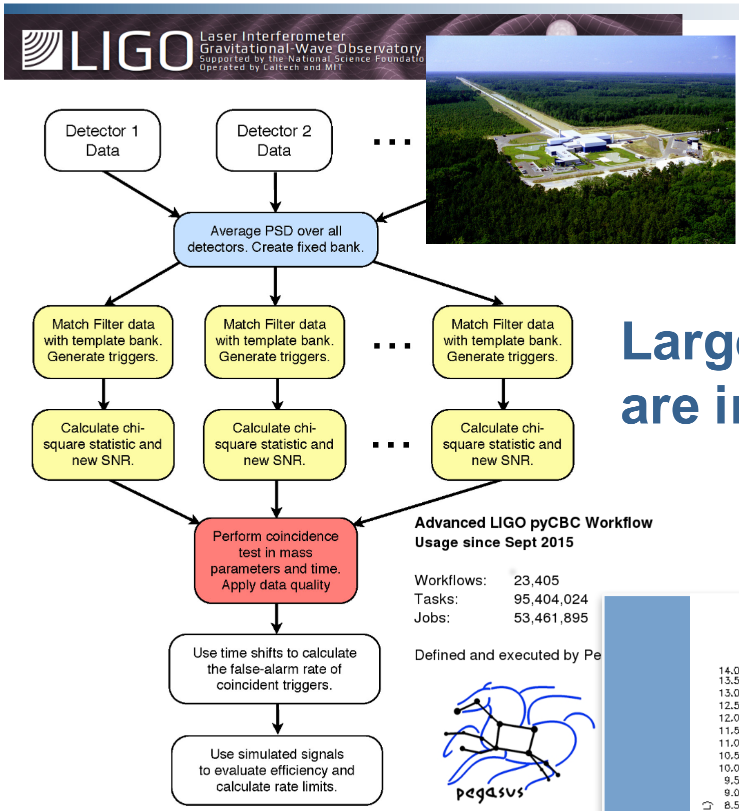
Incoherent

Post-processing and Viz

What we learned in distributed area WMS
We can apply In-Situ workflow management

- **Provision HPC resources ahead of execution:**
  - Keep the resources for multiple tasks, exploit data locality
  - Support provisioning of storage/ incl. NVRAM, burst buffers
  - Alternatively explore interfaces between the WMS and the scheduler, support data-aware scheduling
- **Reliability: WMS deal with: task failures, problems accessing data, resource failures, and others.**
  - Investigate how data replication techniques can be used to improve fault tolerance, while minimizing the impact of energy consumption
  - Explore tradeoffs between data re-computation and data retrieval from DRAM/NVRAM/disk (time to solution and energy consumption)
- **Provenance Capture and Reproducibility: WMS capture provenance information about the creation, planning, and execution**
  - Provenance capture may need to adapt to the behavior of the application (coarse and fine levels of details, compression)
  - May want to automatically re-run parts of the computation and re-produce the results and a more detailed provenance trail on demand

Developing solutions that impact science


LIGO Laser Interferometer Gravitational-Wave Observatory
Supported by the National Science Foundation
Operated by Caltech and MIT
pegasus

Detector 1 Data

Detector 2 Data

. . .

Average PSD over all detectors. Create fixed bank.

Match Filter data with template bank. Generate triggers.

Match Filter data with template bank. Generate triggers.

. . .

Match Filter data with template bank. Generate triggers.

Calculate chi-square statistic and new SNR.

Calculate chi-square statistic and new SNR.

. . .

Calculate chi-square statistic and new SNR.

Perform coincidence test in mass parameters and time. Apply data quality

Use time shifts to calculate the false-alarm rate of coincident triggers.

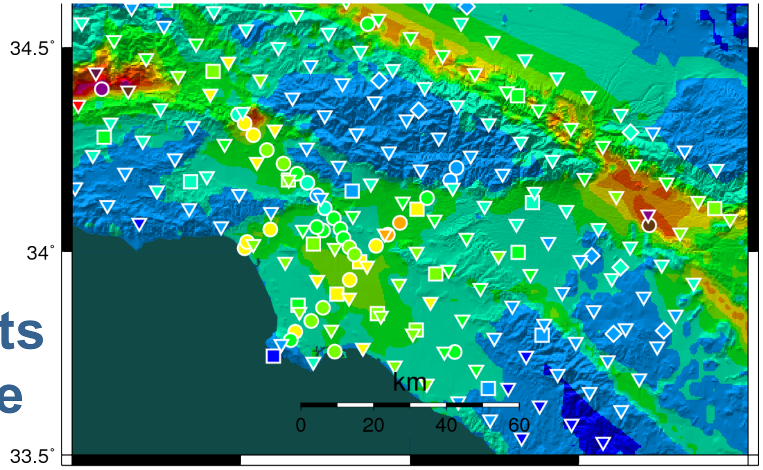
Use simulated signals to evaluate efficiency and calculate rate limits.

Advanced LIGO pyCBC Workflow
Usage since Sept 2015

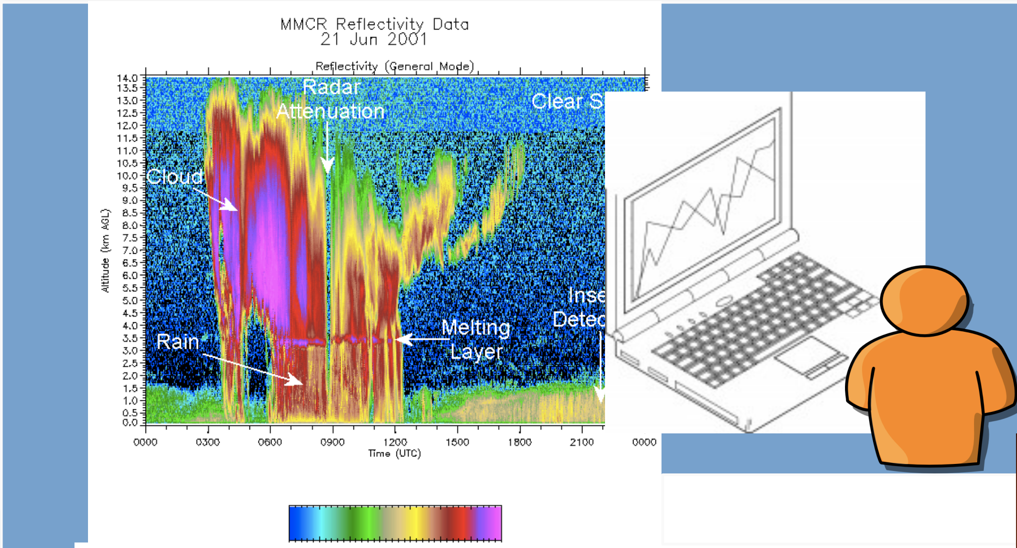
Workflows:	23,405
Tasks:	95,404,024
Jobs:	53,461,895

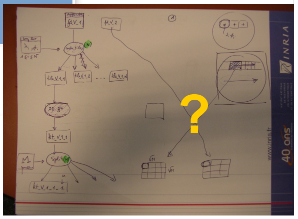
Defined and executed by Pe

Large scale projects are in decent shape



Southern California Earthquake Center





Individuals are still struggling

http://www.teachingboxes.org/avc/content/Cloud_Radar.htm

USC Viterbi School of Engineering