USC Viterbi
School of Engineering
Information Sciences Institute

The Pegasus Workflow Management System: Current Applications and Future Directions

Ewa Deelman, Ph.D.

University of Southern California,
Information Sciences Institute

Multicore 2020 Invited presentation
Wellington, NZ, February 19, 2020

USC University of Southern California

Oldest private university in western U.S 1880.

Diverse student population

18 professional schools

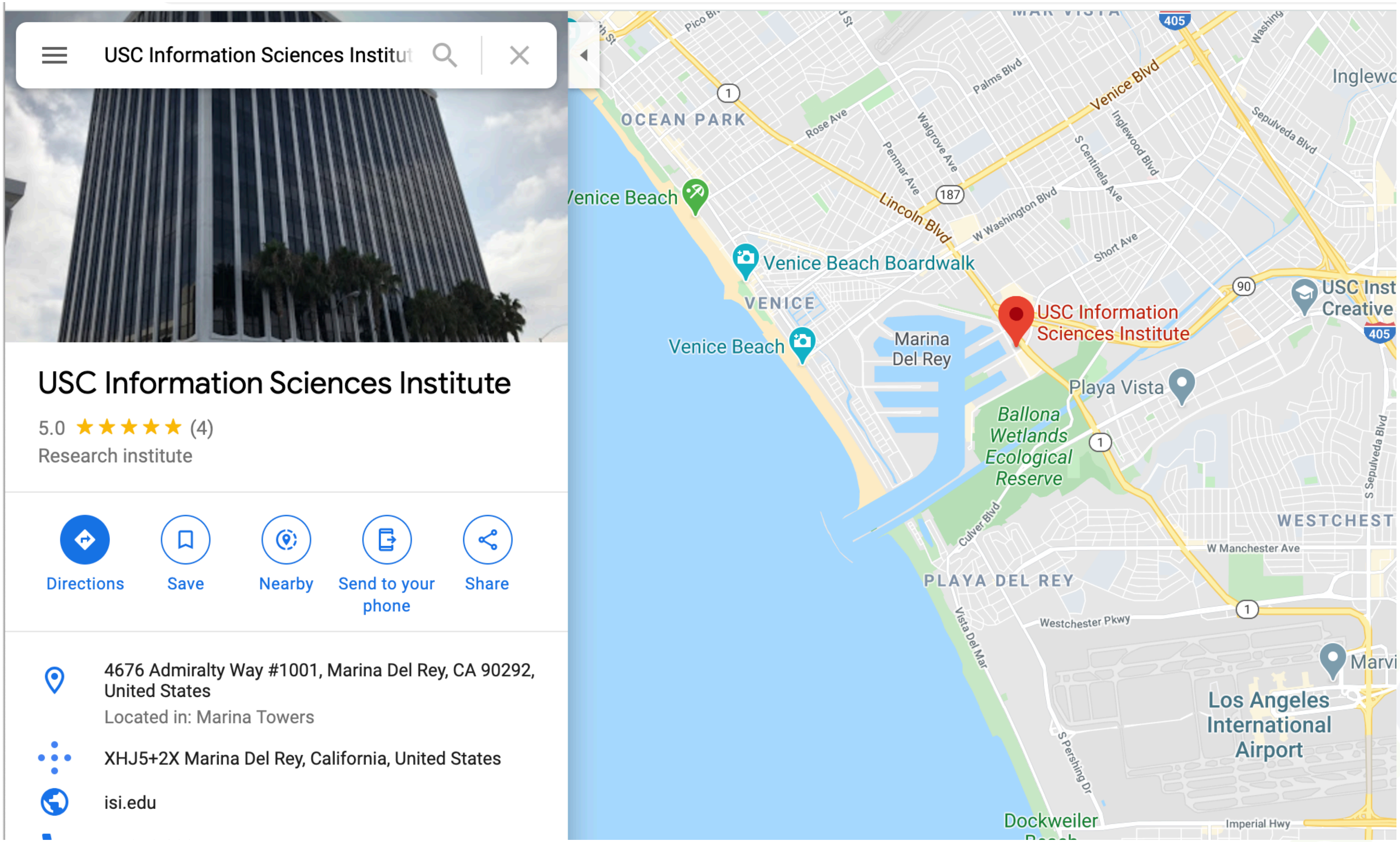
Located in Los Angeles, CA

Students (2019-2020 academic year)

Rounded to the nearest 500

Undergraduates	20,500
Graduate and professional	28,000
Total	48,500

USC Information Sciences Institute, near Los Angeles, CA



USC Information Sciences Institute

USC Viterbi School of Engineering Information Sciences Institute

A unit of USC's Viterbi School of Engineering

- Principally located in USC's Marina Tech Campus in Marina del Rey, CA

400 faculty, staff, and students (June 2019)

- Affiliations and associations with multiple USC engineering departments, additional schools

$110M/year external funding support (2018) from a diverse base of research sponsors

- DARPA, IARPA, NIH, NSF, DOE, …

Research Areas:

Advanced electronics

–MOSIS low-cost prototype and small-volume chip fabrication; novel electronics

Computational systems and technology

–Software/hardware supercomputing, high-performance computing, **distributed computing, scientific workflows**

Informatics

–Medical informatics, decision systems, computer networks, grid computing

Intelligent systems / artificial intelligence

–Natural language, knowledge technologies, information and geospatial integration, robotics

Networking and cybersecurity

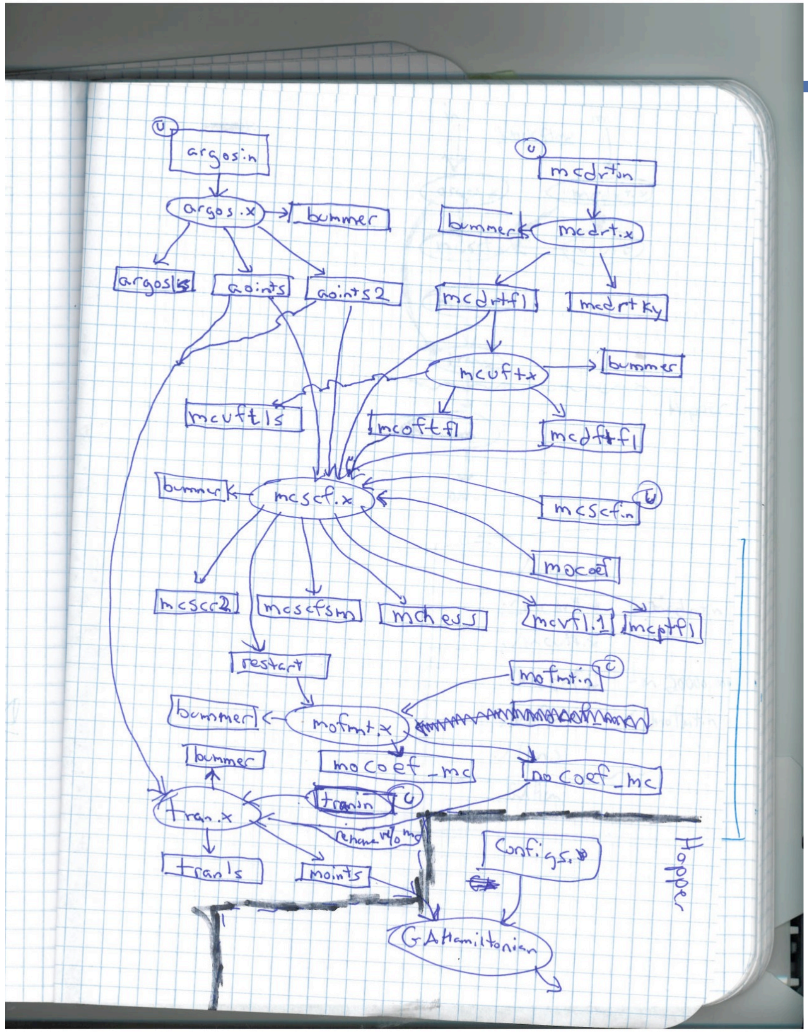
Benefits of Scientific Workflows (from the point of view of an application scientist)

USC Viterbi School of Engineering
Information Sciences Institute

- Conducts a series of computational tasks.
  - Resources distributed across Internet.
- Chaining (outputs become inputs) replaces manual hand-offs.
  - Accelerated creation of products.
- Ease of use - gives non-developers access to sophisticated codes.
  - Avoids need to download-install-learn how to use someone else's code.
- Provides framework to host or assemble community set of applications.
  - Honors original codes. Allows for heterogeneous coding styles.
- Framework to define common formats or standards when useful.
  - Promotes exchange of data, products, codes. Community metadata.
- Multi-disciplinary workflows can promote even broader collaborations.
  - E.g., ground motions fed into simulation of building shaking.
- Certain rules or guidelines make it easier to add a code into a workflow.

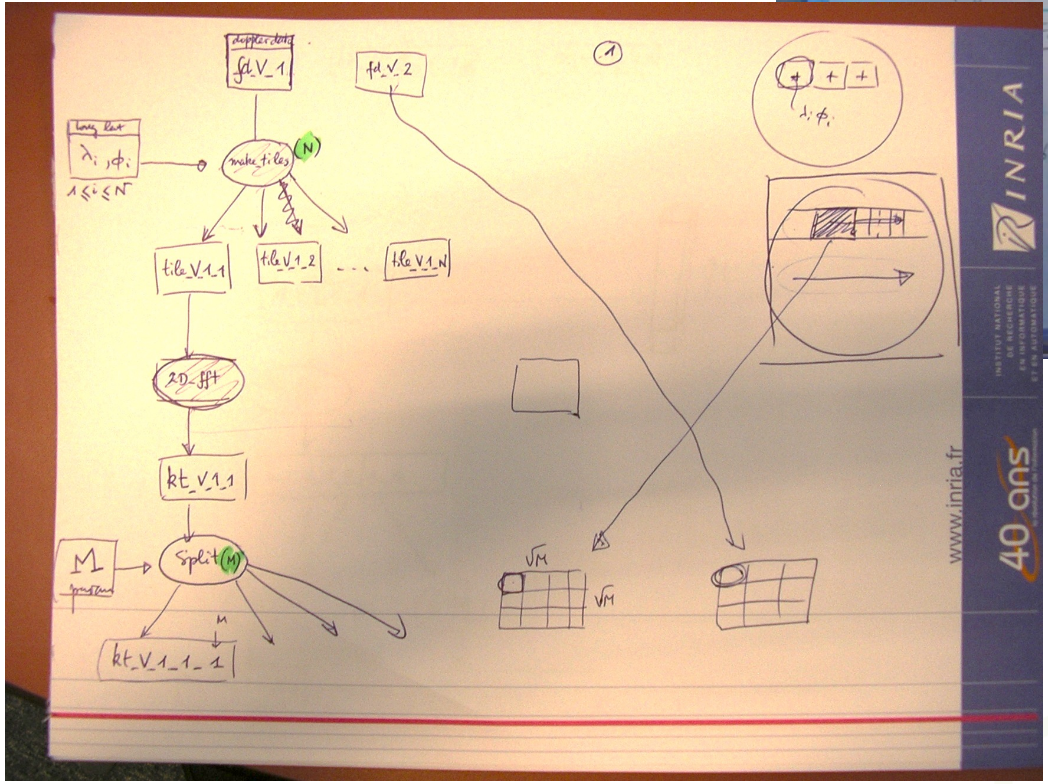
Slide courtesy of David Okaya, SCEC, USC

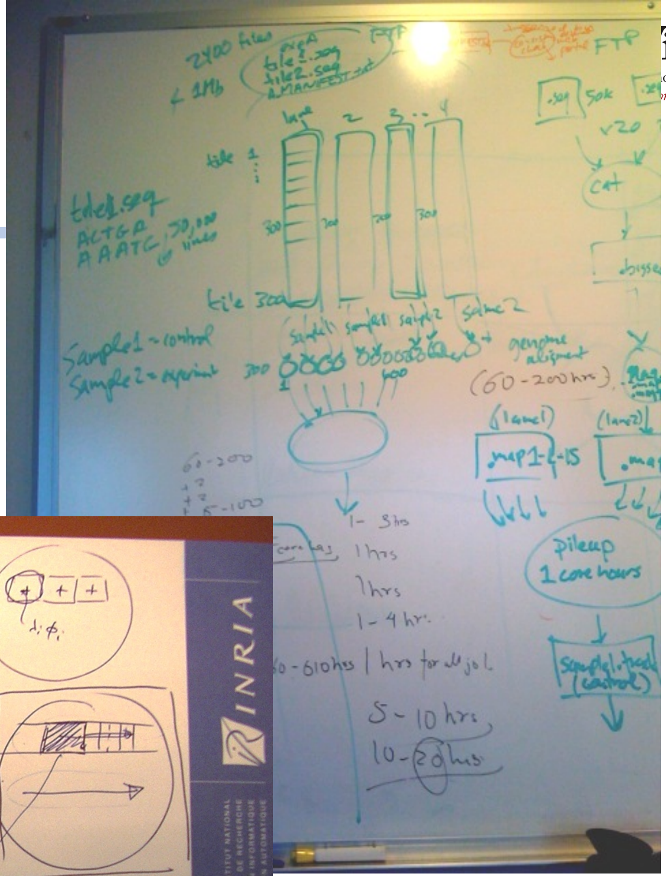
How do workflows start?



Solar fuels: integrate light absorption and electron transfer driven catalysis, RENCI

Helioseismology, Max Planck





RNA sequencing, USC

Workflows Live in a Heterogeneous World

- Data sources (files) are distributed
  - "Same" data can be replicated
  - Data access protocols are heterogeneous, authentication methods are heterogeneous
  - Data access varies by location and time period
  - Need to know the data name, location, data transfer protocol
- Computational platforms are heterogeneous
  - HPC, HTC, clouds and everything in between
  - Different authentication mechanisms, different job scheduling interfaces
  - Need to know system characteristics, scheduling interfaces
- Can also configure networks via SDN

Challenges of Workflow Management

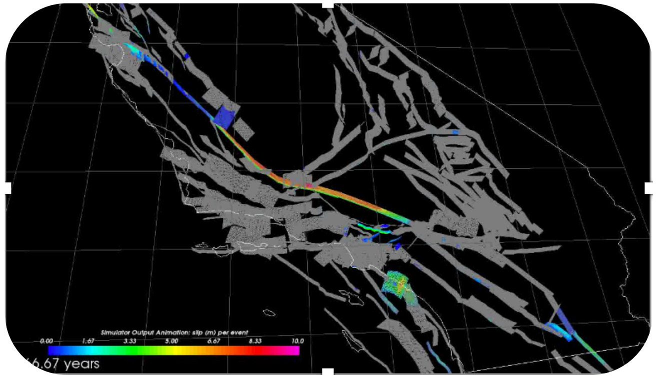
Challenges across domains

- Need to describe complex workflows in a simple way
- Need to access distributed, heterogeneous data and resources (heterogeneous interfaces)
- Need to deal with resources/software that change over time

Our focus

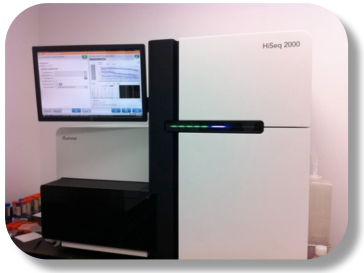
- Separation between workflow description and workflow execution
- Workflow planning and scheduling (scalability, performance)
- Task execution (monitoring, fault tolerance, debugging)

Sky mosaic, IPAC, Caltech



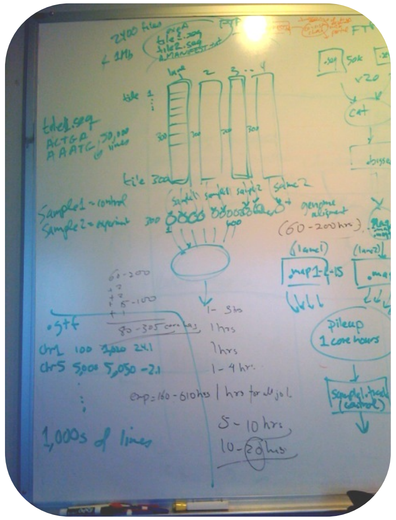
Earthquake simulation, SCEC, USC

Submit locally run globally



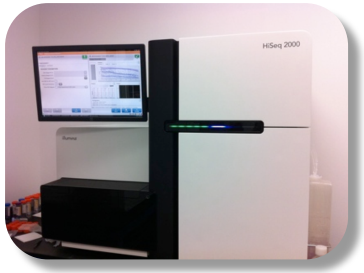
Local Data Storage

Work Definition



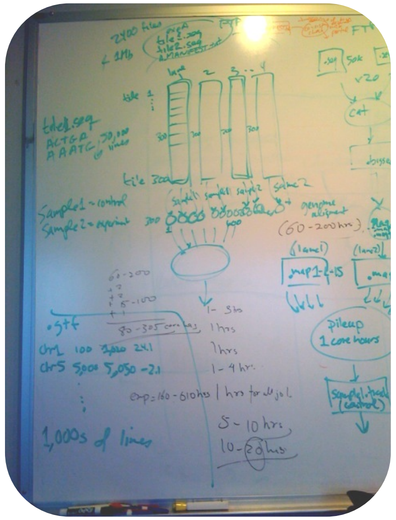
Local Resource

Typical Local Computational Environment



Local Data Storage

Work Definition



Local Resource

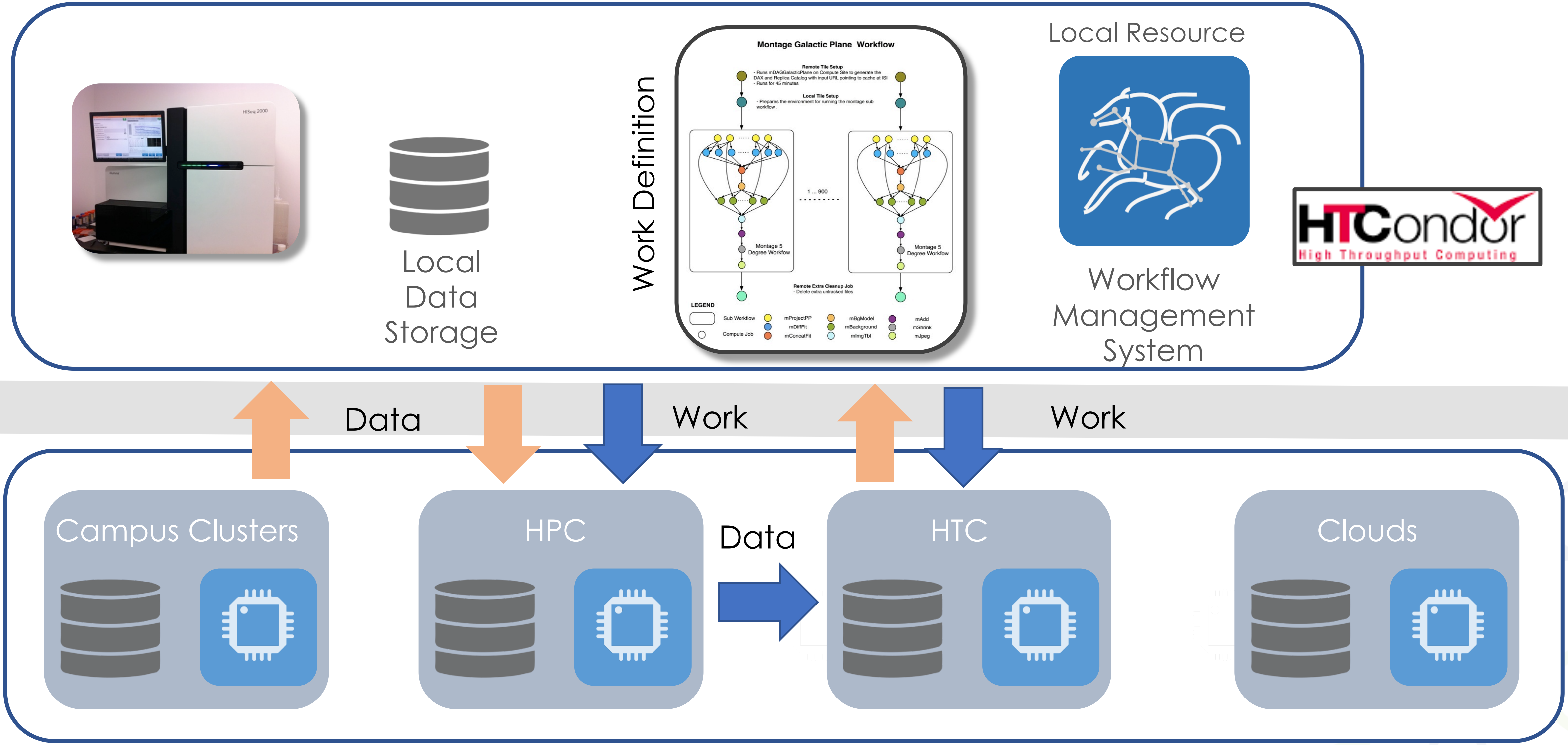
Campus Clusters

HPC

HTC

Clouds

Submit locally run globally



USC Viterbi
School of Engineering
Information Sciences Institute

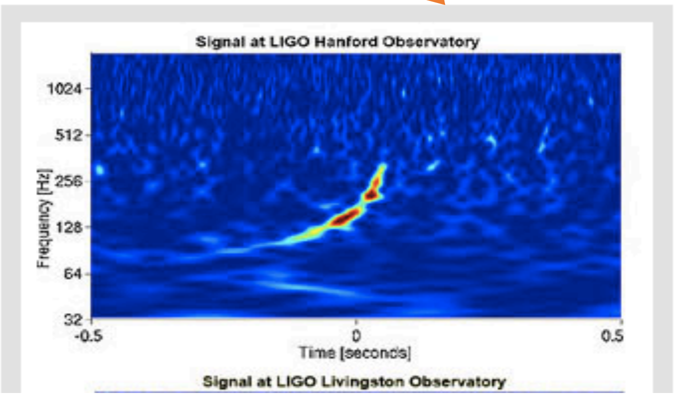
Pegasus: Grounding Research and Development

Nobel Prize

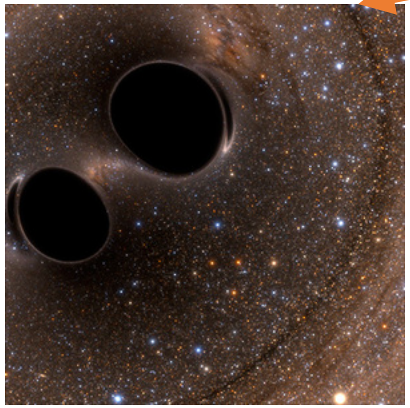
Working with LIGO

2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017

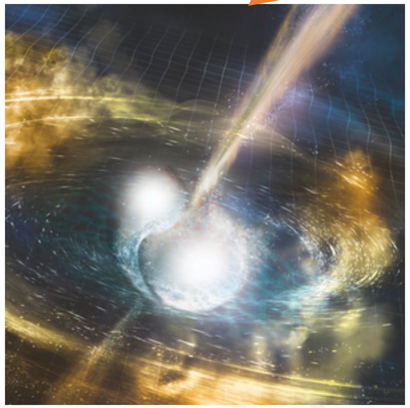
First Pegasus prototype



Blind injection detection



First detection of black hole collision



Multi-messenger neutron star merger observation

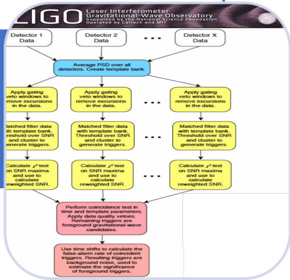
Image credit: LIGO Scientific Collaboration

Complexity of LIGO Workflows

First GW detection: ~ 21K Pegasus workflows, ~ 107M tasks

Analysis measures the statistical significance of collected data

Science Workflow



Efficient, scalable, and robust execution of tasks and data access

Automation

pegasus

LIGO, Open Science Grid, XSEDE, Blue Waters

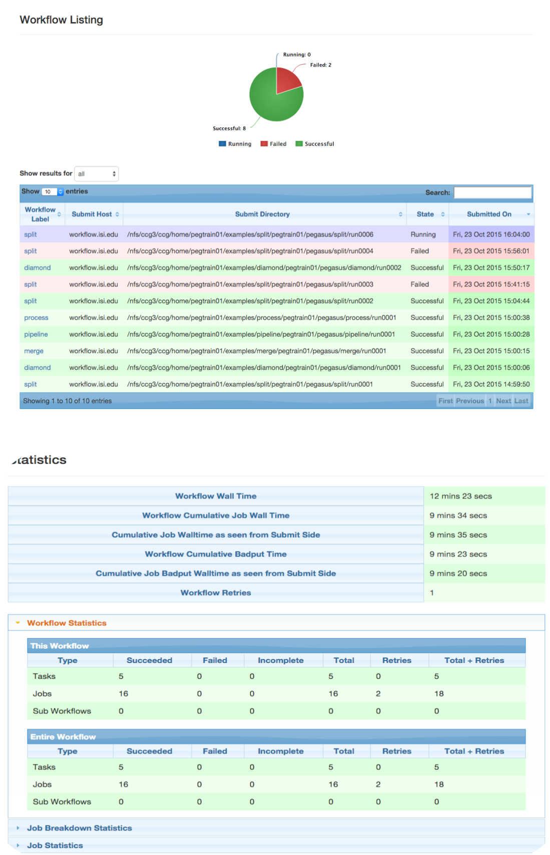
Distributed Power

2015/16

USC Viterbi
School of Engineering
Information Sciences Institute

Pegasus Workflow Management System

- Operates at the level of files and individual applications
- Allows scientists to describe their computational processes (workflows) at a logical level
- Without including details of target heterogeneous CI (portability)
- Scalable to O(10^6) tasks, TBs of data
- Captures provenance and supports reproducibility
- Includes monitoring and debugging tools



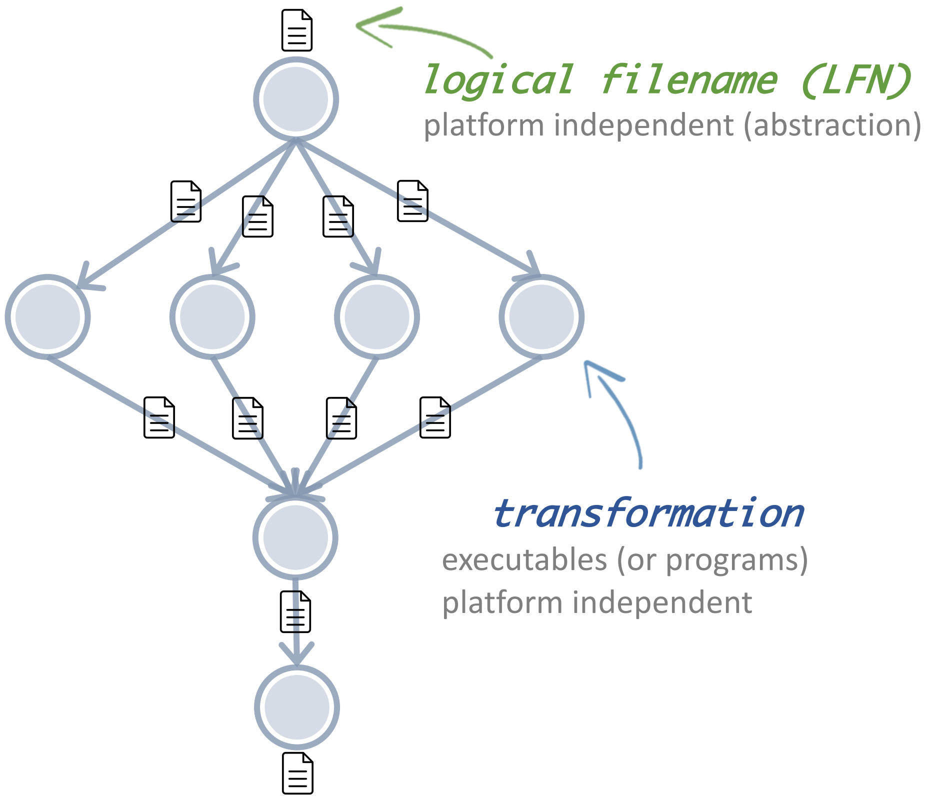
Composition in Python, R, Java, Perl, Jupyter Notebook

hubzero

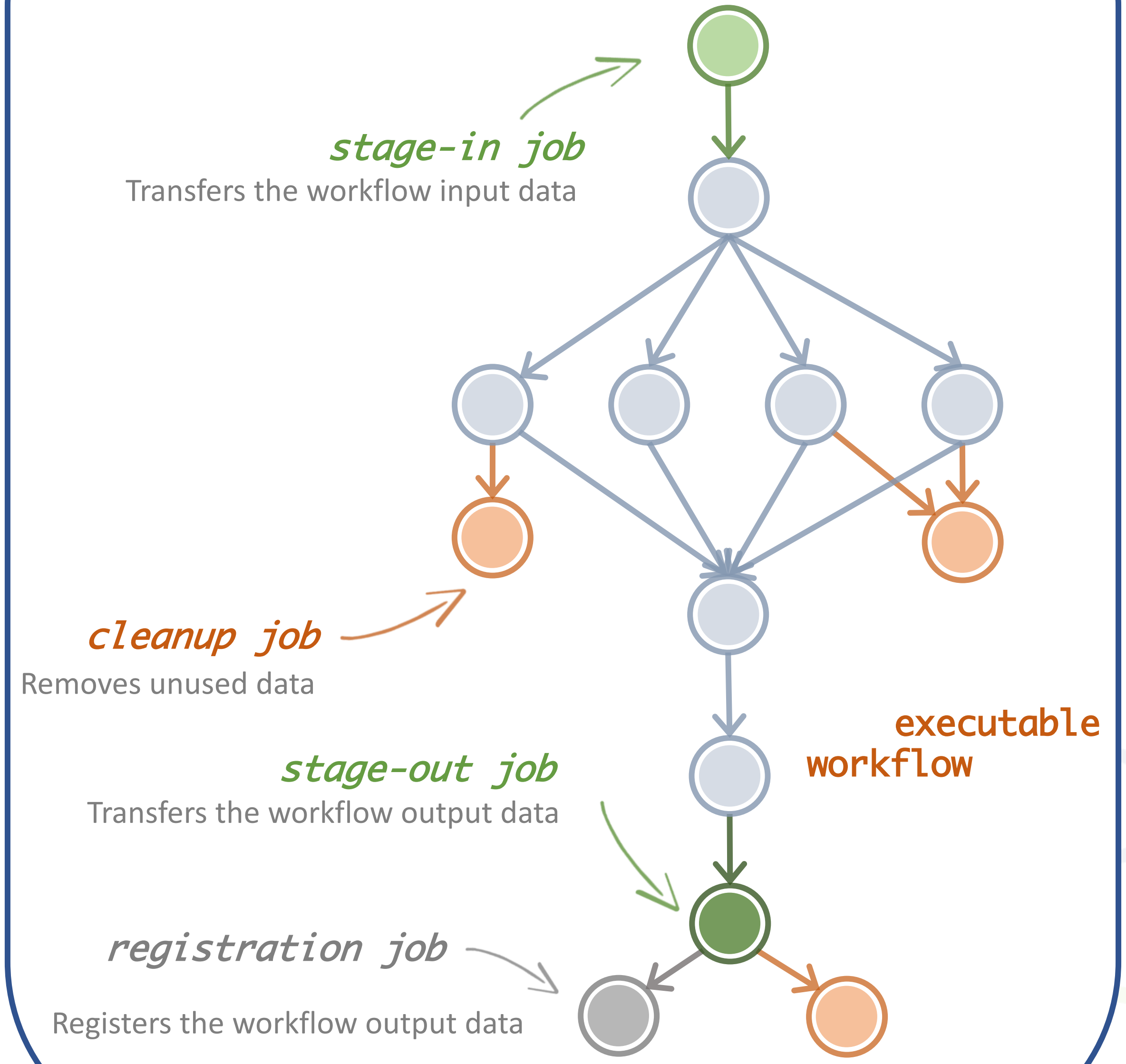
Abstract Workflow

Portable Description

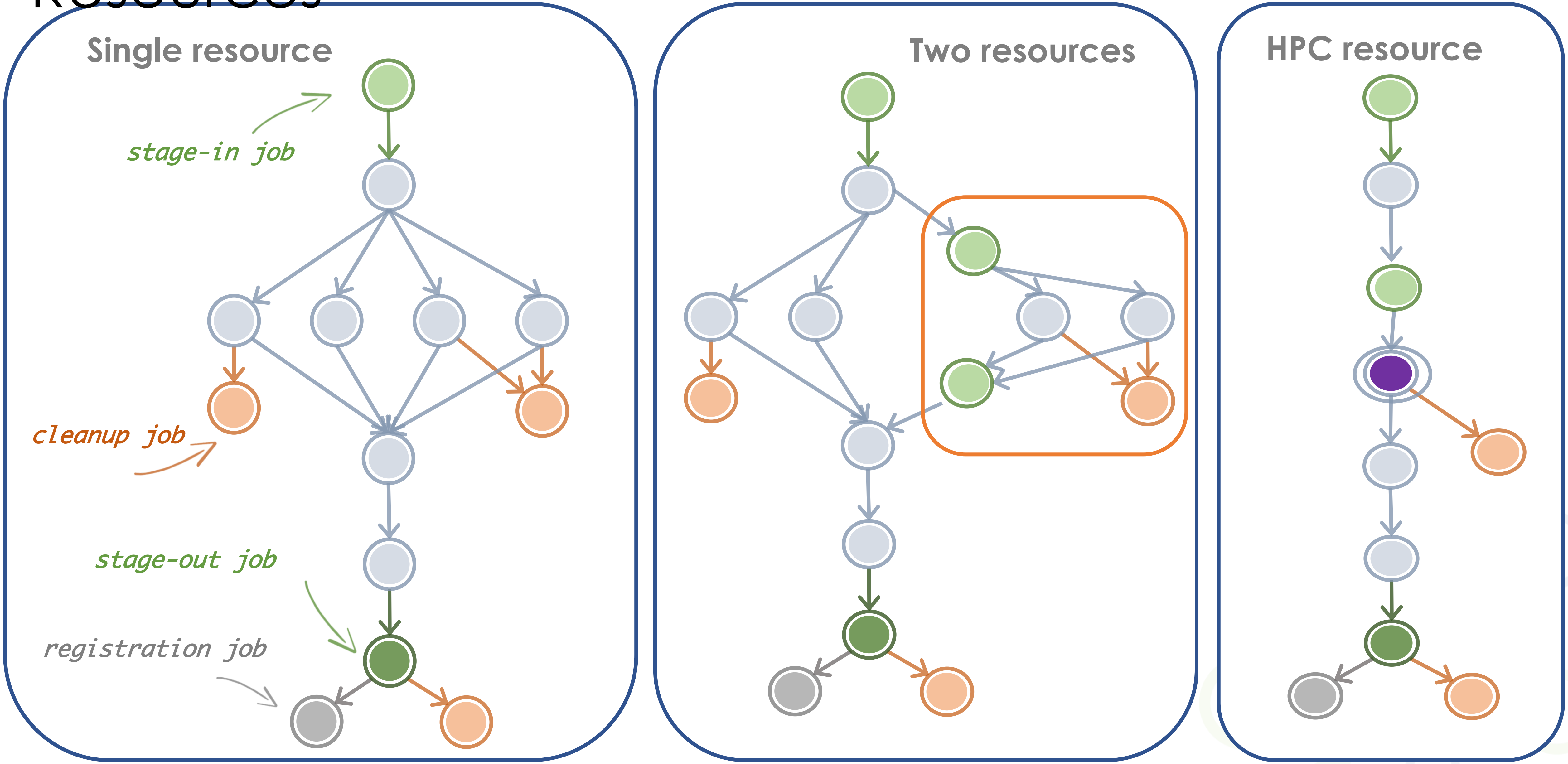
Users do not worry about
low level execution details



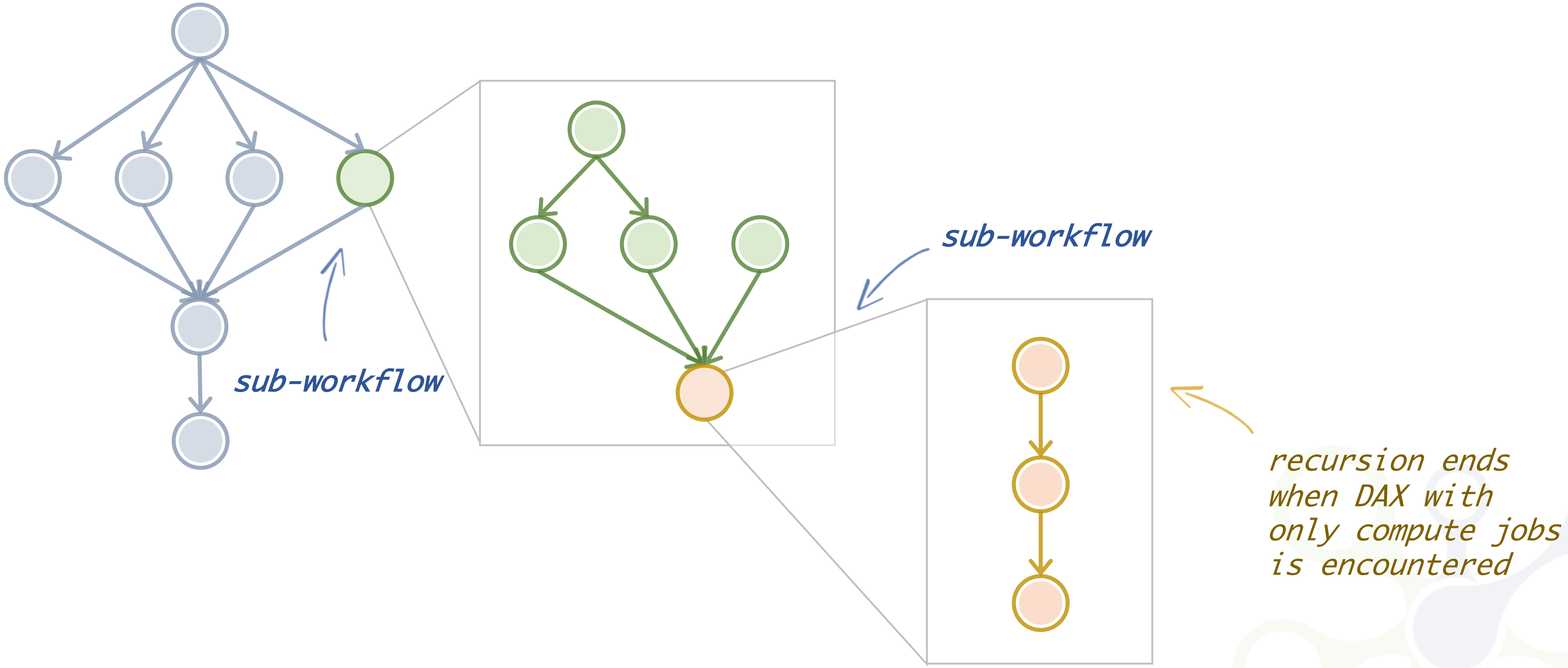
Executable Workflow



Executing Workflows on Heterogeneous Resources

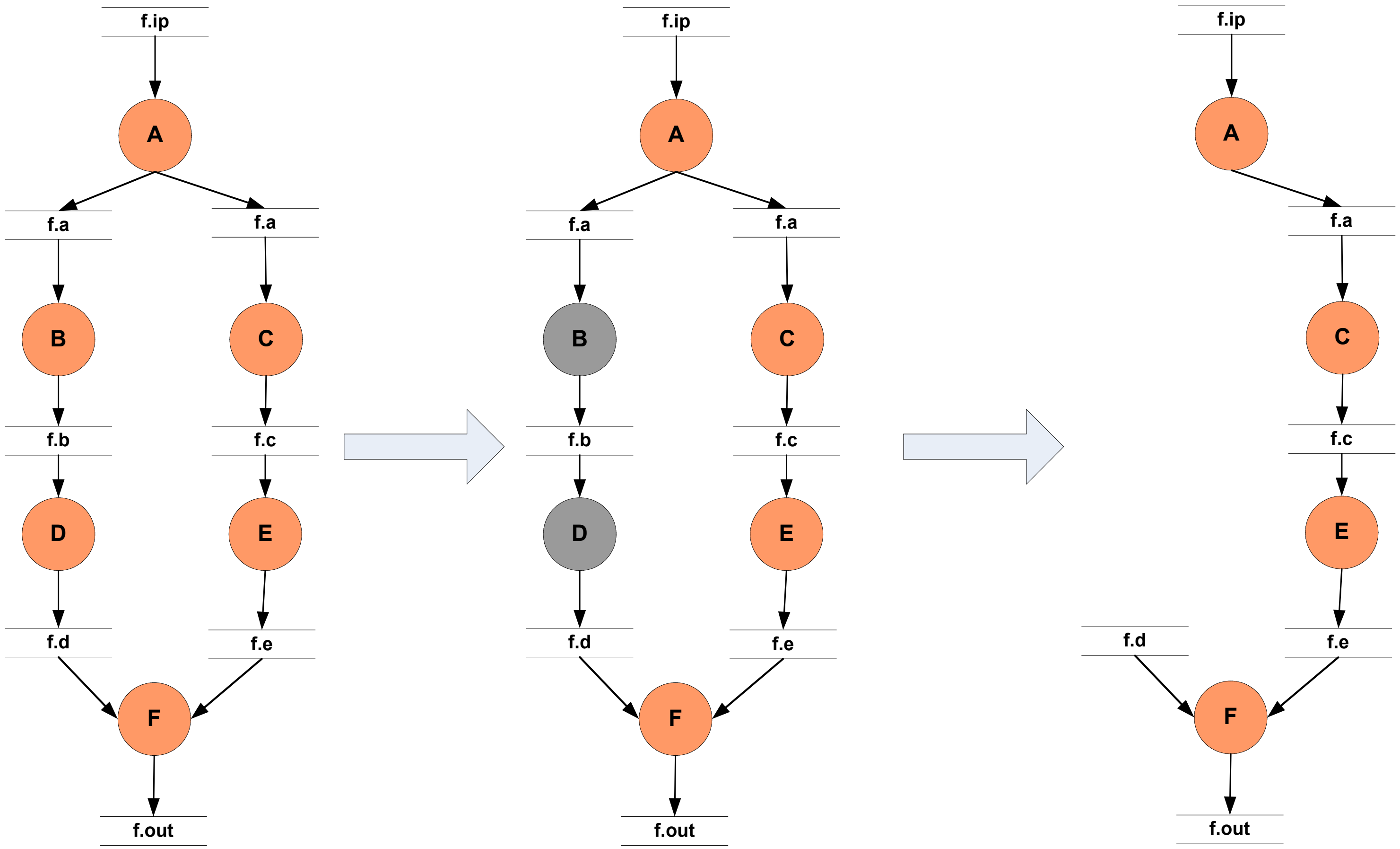


Managing Large-Scale Workflows



Data Re-use and Resilience

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



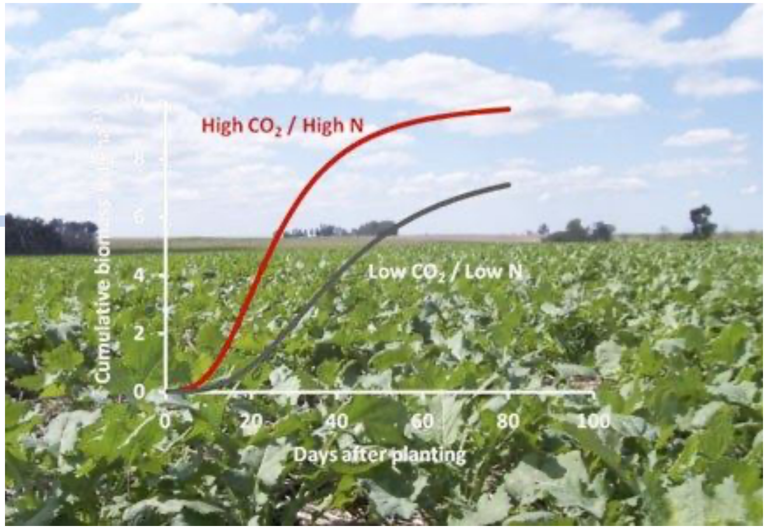
Abstract Workflow

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*

Example application: Crop modeling



Computational science in the **long tail** still imposes challenges when, for example, addressing complex research questions:

How the average crop production would respond to different levels of soil fertilization?

Which fraction of weed would harm crop production for different crops in distinct regions?

How do fertilizer level and weed pressure interact across regions, years or planting dates within a growing season?

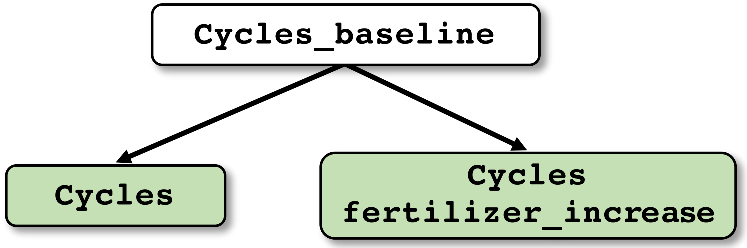
mint
model integration

Armen R. Kemanian, Penn State

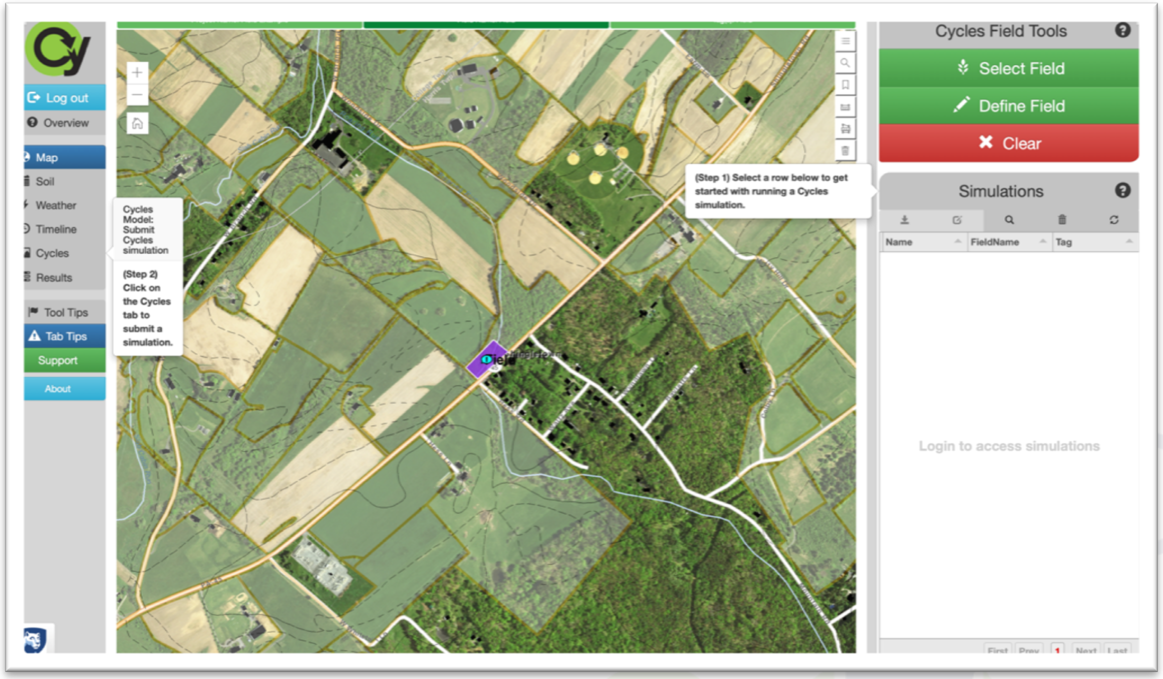
The Cycles HTC Workflow

User-friendly, multi-crop, multi-year, process-based **Agroecosystem model** with daily time step simulations of **crop production** and the water, carbon (C) and nitrogen (N) cycles in the soil-plant-atmosphere continuum

https://psumodeling.github.io/Cycles/



Overview of a single configuration definition of a Cycles abstract (i.e., platform and scenario agnostic) workflow. The model output is evaluated against a reference or baseline.



Cycles Simulation Matrix

Parameter	Values
Country	South Sudan
Crop	Maize, Sorghum, Sesame, Peanut
Start planting date	100, 107, 114, 121, 128, 135, 142
End planting date	149
Planting date fixed	True, False
Nitrogen rate	0, 25, 50, 100, 200, 400
Weed fraction	0.0, 0.05, 0.1, 0.2, 0.4, 1.5, 2.0

A cross-product of this use-case parameters generates over **1.8M tasks**

Input data:

- weather data for inferring weather conditions such as rain, air temperature, wind
- ~102GB sized files , about 6,500 files (one file per day for years between 2000-2017)

Rafael Ferreira Da Silva, USC/ISI

Execution Profiles of the Cycles Workflow Jobs

very short-running tasks

Task	Count	Runtime μ	Runtime σ	CPU Utilization μ	CPU Utilization σ	I/O Read μ	I/O Read σ	I/O Write μ	I/O Write σ
GLDAS_to_Cycles	209	22,486.3	142.7	84.1%	1.2%	1,001,690	0.0	0.5	0.0
Cycles_baseline	614,460	8.2	3.1	75.2%	1.4%	0.5	0.0	21.4	0.1
Cycles_fertilizer_increase	614,460	6.1	2.3	65.6%	1.8%	0.5	0.1	21.6	0.1
Cycles	614,460	6.2	2.4	59.7%	2.1%	0.5	0.1	21.6	0.1
Cycles_output_summary	1,045	3.6	1.1	72.3%	2.4%	4,514.4	0.1	2.7	0.0
generate_graphs	1	20.3	–	81.2%	–	2,821.5	–	11.3	–

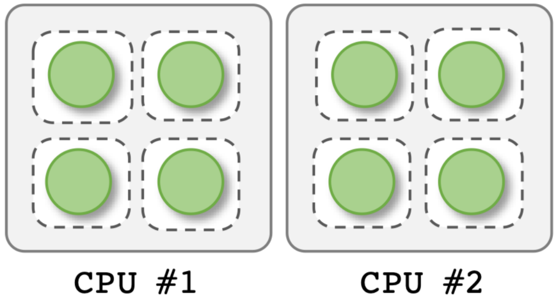
RUNTIMES ARE SHOWN IN SECONDS, AND I/O OPERATIONS IN MB. (μ IS THE MEAN, AND σ THE STANDARD DEVIATION.)

Overhead (e.g., queuing time) for short-running tasks may be counter-productive

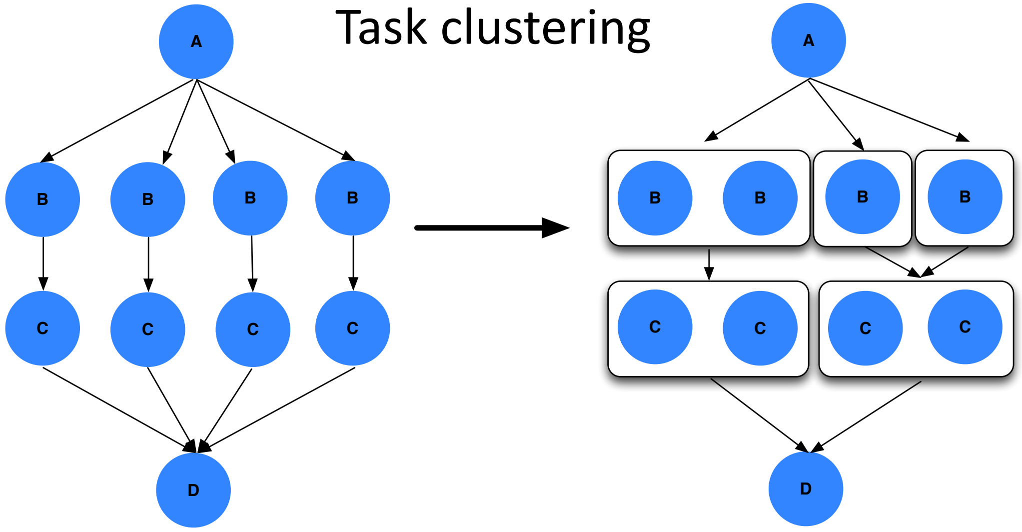
Pegasus Optimizations

well-known optimizations

Task-resource co-allocation



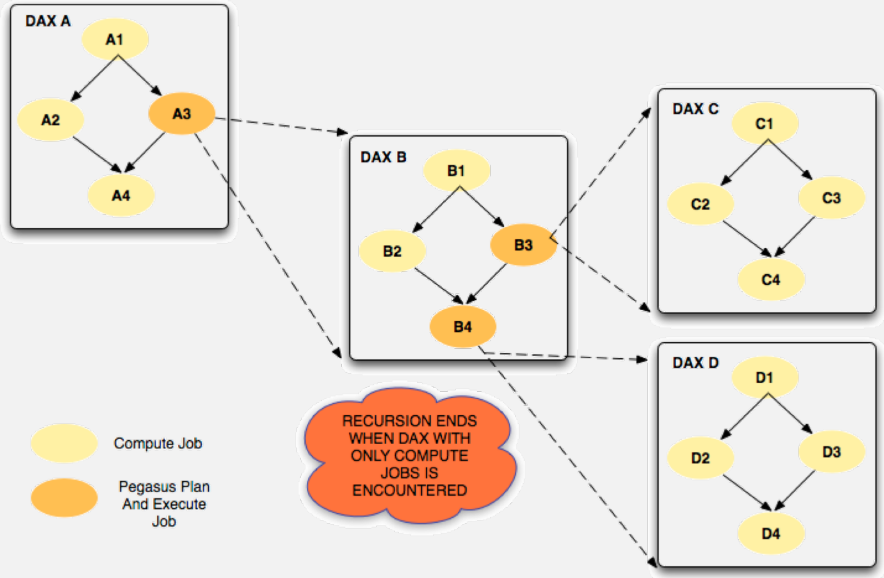
Task clustering



Hierarchical workflows

Enacts the execution of **millions of tasks**

Also enables loops and conditionals in DAGs



Containers

Fosters **reproducibility** and mitigates software dependencies issues

modern workflow optimizations

Experimental Results

Method	Cycles_baseline 10	15	20	25	50	Cycles_fertilizer_increase 10	15	20	25	50	Cycles 10	15	20	25	50
1 task per core	93.7 (±10.3)	138.6 (±18.2)	186.2 (±22.0)	219.8 (±22.4)	391.6 (±52.5)	61.9 (±5.6)	93.5 (±10.0)	124.4 (±14.3)	161.9 (±16.4)	355.2 (±72.6)	62.5 (±6.1)	93.6 (±8.3)	125.8 (±14.4)	163.8 (±14.9)	364.1 (±70.7)
Co-allocation	99.2 (±10.1)	141.7 (±14.3)	196.9 (±19.2)	242.3 (±30.2)	557.9 (±84.3)	64.3 (±7.3)	100.0 (±7.9)	129.9 (±19.9)	163.3 (±11.5)	360.1 (±46.5)	64.5 (±7.4)	100.3 (±10.1)	136.4 (±13.4)	165.4 (±14.5)	371.9 (±49.4)

2 tasks per core

RUNTIME IN SECONDS FOR **CYCLES SIMULATION CLUSTERED JOBS**

Method	No clustering	Cluster Sizes 10	15	20	25	50
1 task per core	311.1	181.2	128.7	105.2	105.1	104.9
Co-allocation	237.4	128.9	114.1	99.2	93.8	86.6

WORKFLOW MAKESPAN IN HOURS

**CVMFS

Improves I/O throughput by constantly avoiding pulling the container image from an external endpoint

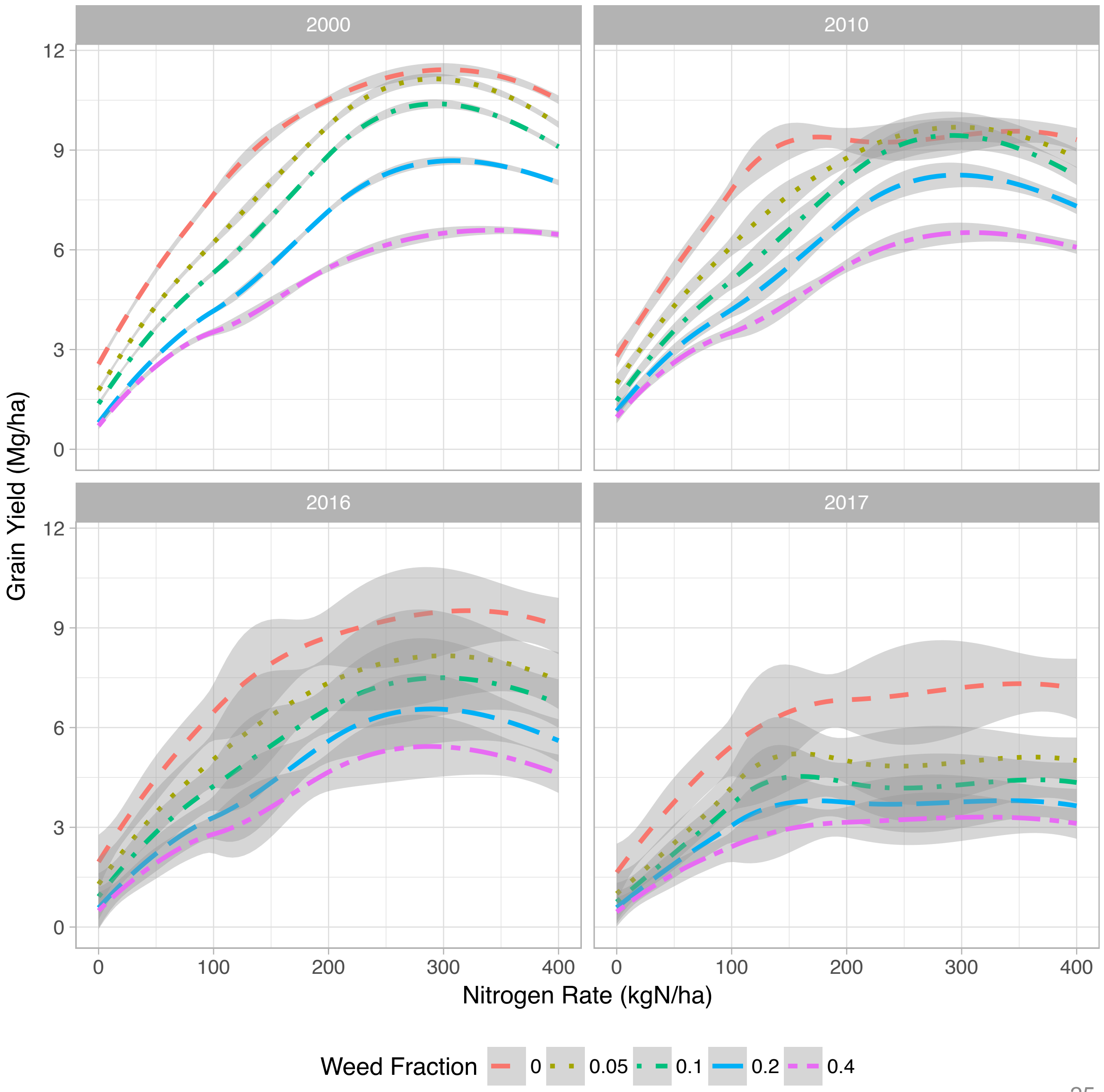
Improvements ~ 3X

Output Products

Simulated grain yield response of **maize** for different fertilization rates and weed pressure levels for a single grid cell

By comparing the simulated grain yield production from 2017 to previous years, it is immediately clear that in that year there was a **steep fall in grain yield** and a weak response to fertilizer

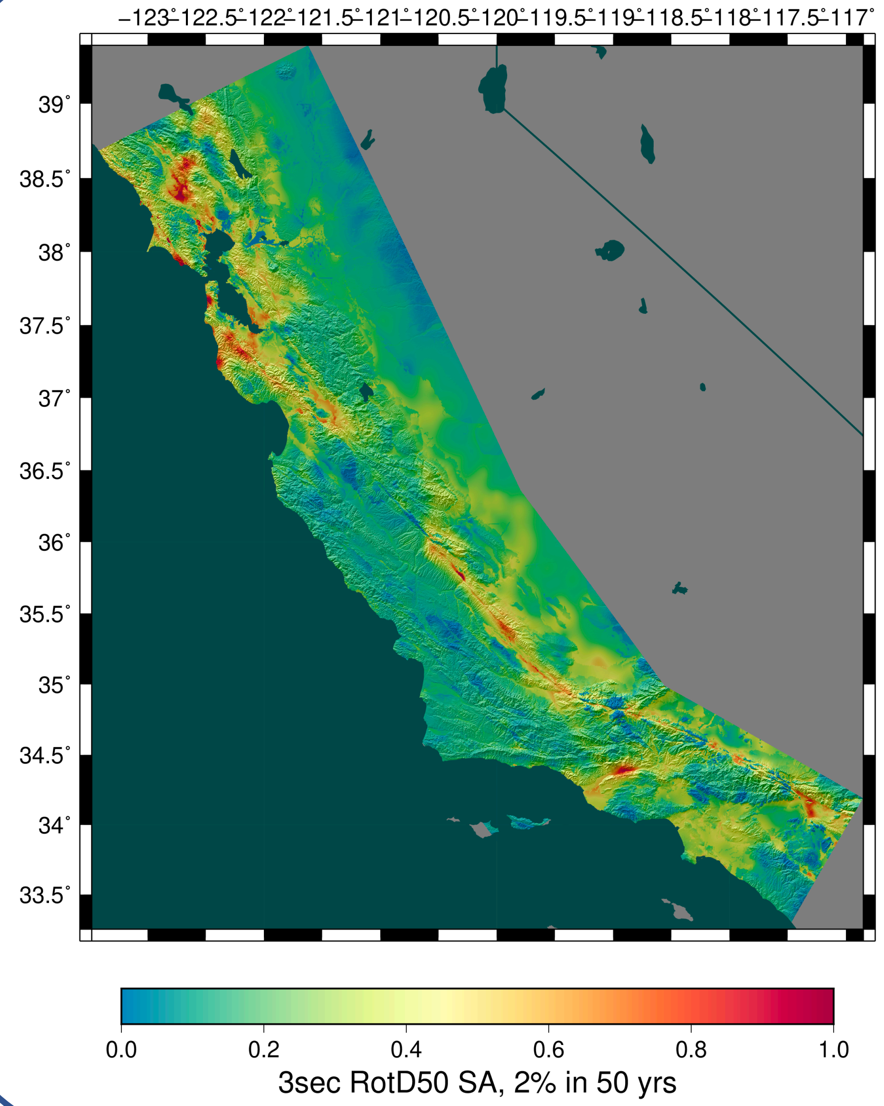
The modeled response to fertilizer is substantial but it depends on weed pressure
Weed control is limited by labor, equipment, or availability of chemicals



Supporting Heterogeneous Workflows

USC Viterbi
School of Engineering
Information Sciences Institute

SCEC's CyberShake: What will the peak earthquake motion be over the next 50 years?



Useful information for:

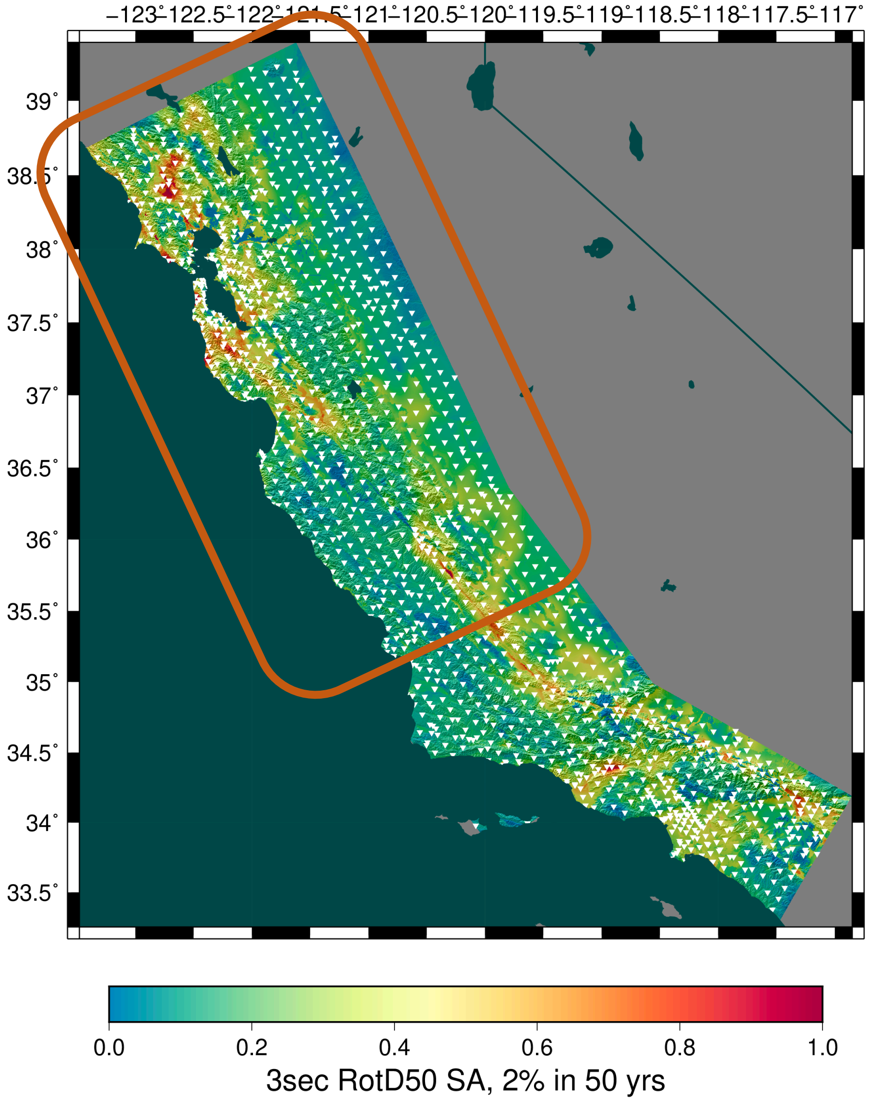
- Building engineers
- Disaster planners
- Insurance agencies

Slide credit: Southern California Earthquake Center

USC Viterbi
School of Engineering
Information Sciences Institute

Supporting Heterogeneous Workflows

2018-2019 Mapping Northern California



- 120 million core-hours
- 39,285 jobs
- 1.2 PB of data managed
- 157 TB of data automatically transferred
- 14.4 TB of output data archived

- NCSA *Blue Waters*
- OLCF *Titan*

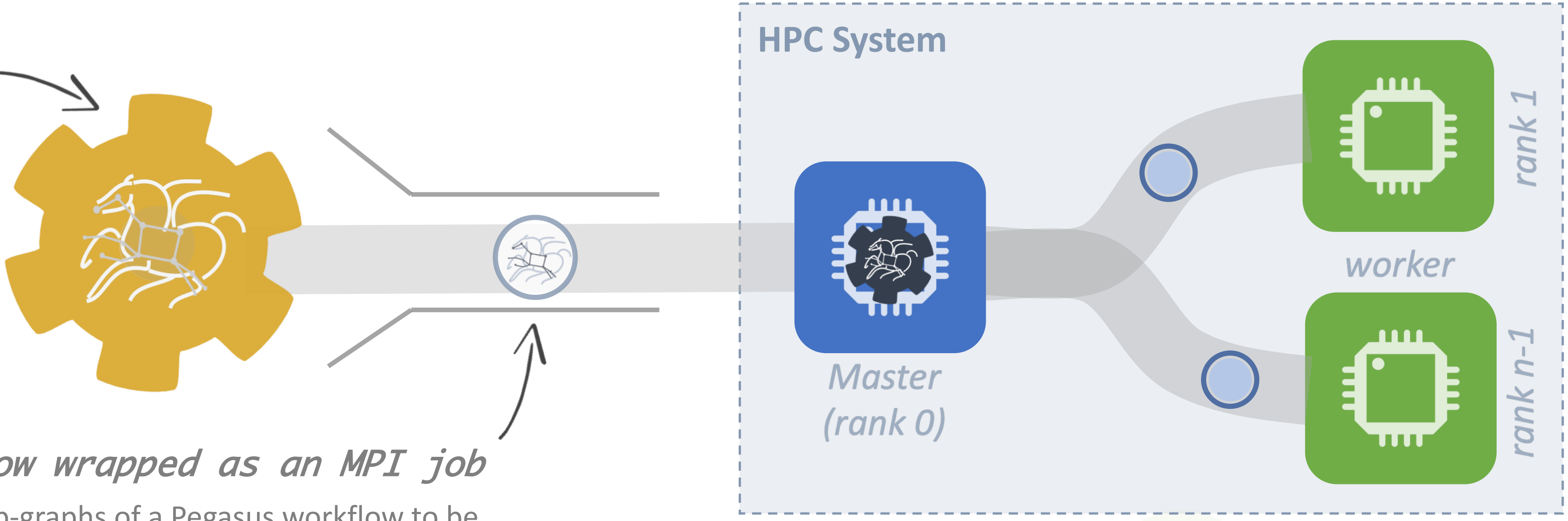
Total map:
170 million core hours
> 19,407 core years

Slide credit: Southern California Earthquake Center

Running **fine-grained** workflows on HPC systems…

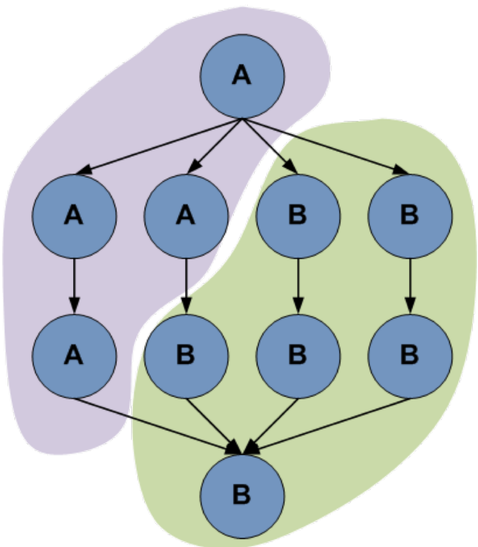
Specialized Workflow Engines Needed for Different Execution Sites

submit host
(e.g., user's laptop)



workflow wrapped as an MPI job

Allows sub-graphs of a Pegasus workflow to be submitted as monolithic jobs to remote resources



Pegasus MPI-Cluster

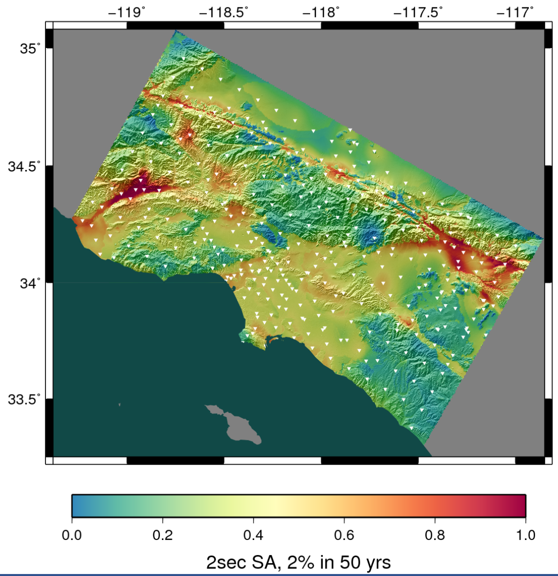
Partition the workflow into sub-workflows and send them for execution to the target system

Mix Workloads on Heterogeneous/ Changing CI

USC Viterbi
School of Engineering
Information Sciences Institute

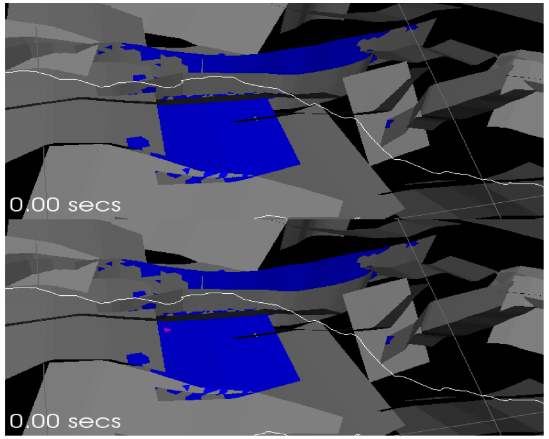
Since 2007: 215 million core-hours (24,543 years)
9 different supercomputers

Pegasus Optimizations:
- Task clustering
- MPI-based workflow engine

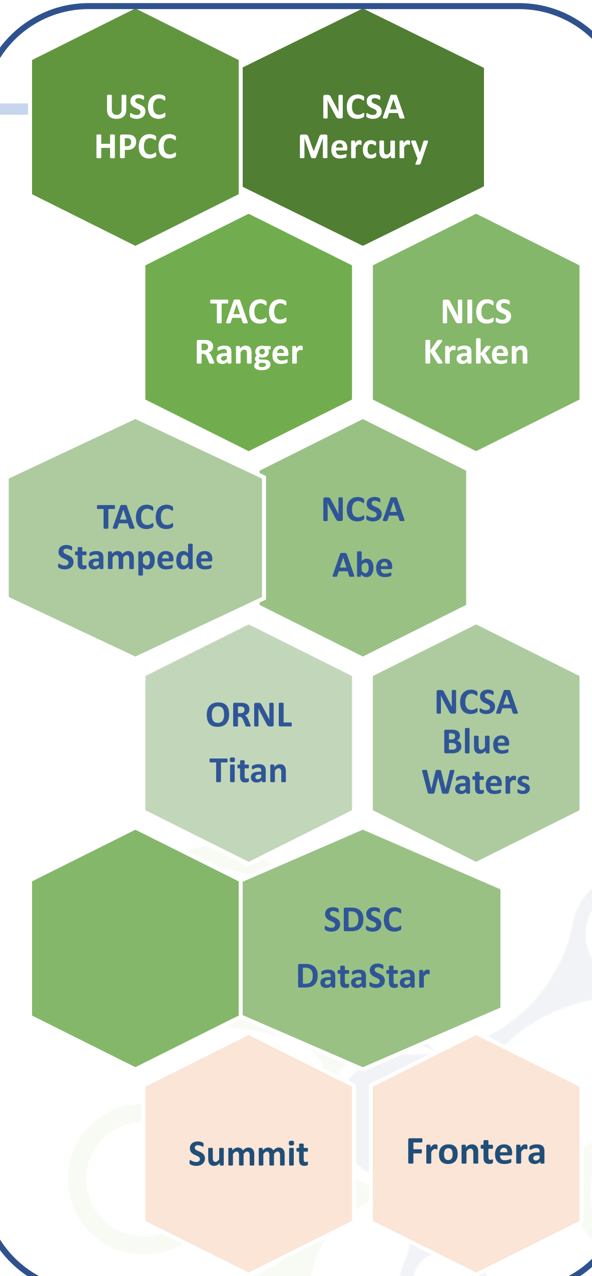


2010: World's first physics-based probabilistic seismic hazard map

Application Optimizations:
- Workflow restructuring
- MPI/code tuning
- Porting to GPUs

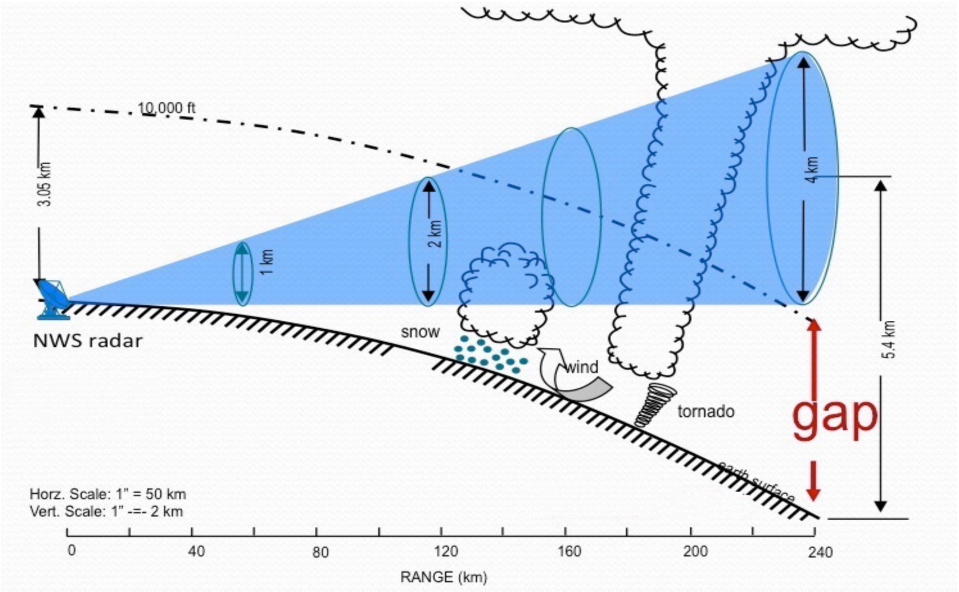


2018: Incorporating earthquake simulator with a 1 million-year catalog of California seismicity

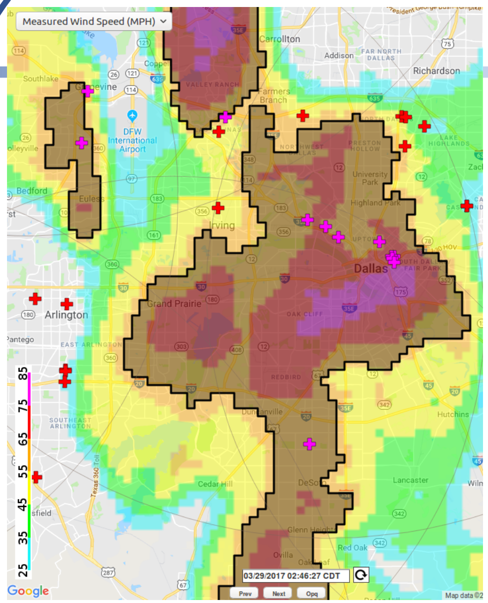


Slide credit: Southern California Earthquake Center

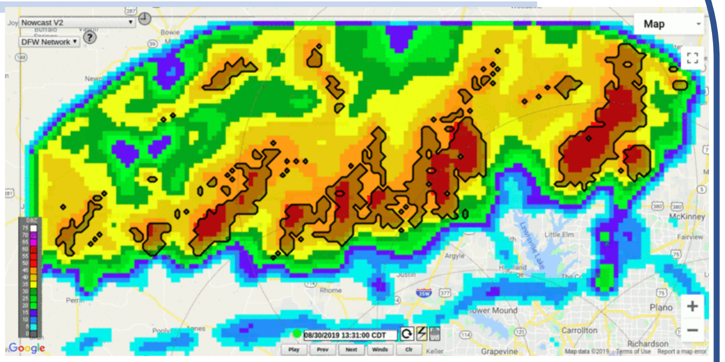
CASA: Collaborative Adaptive Sensing of the Atmosphere



- Network of short range Doppler radars
- Adjustable sensing modes in response to quick weather changes
- Suitable for near-ground weather events: tornado, hail, high winds



Maximum wind velocity
Sends alerts to users



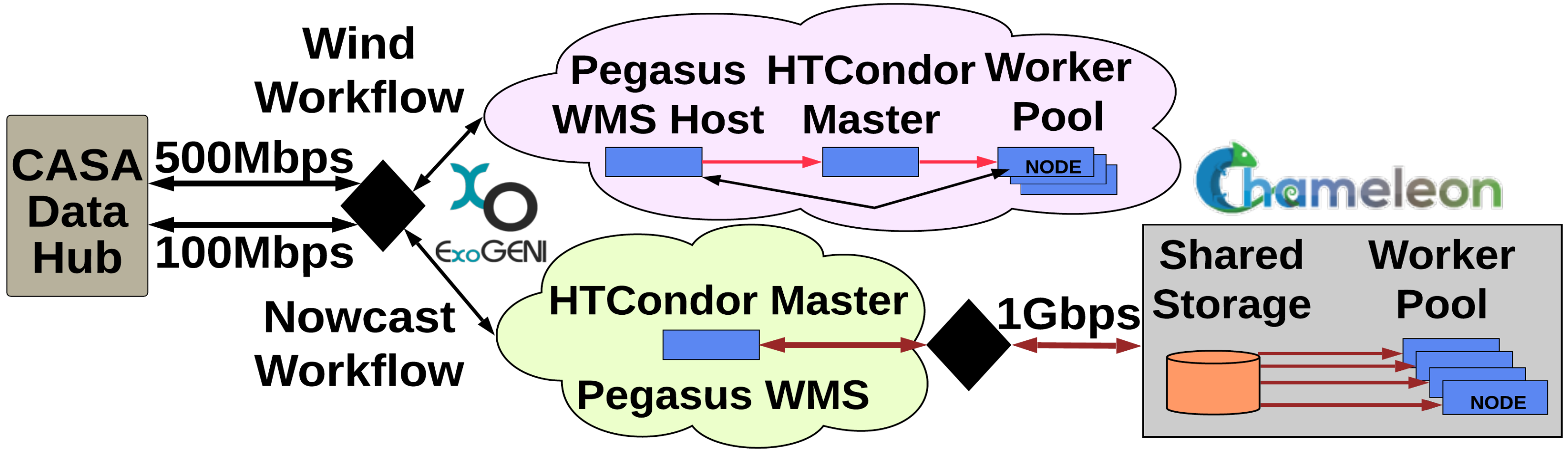
Nowcasts: predict the immediate weather

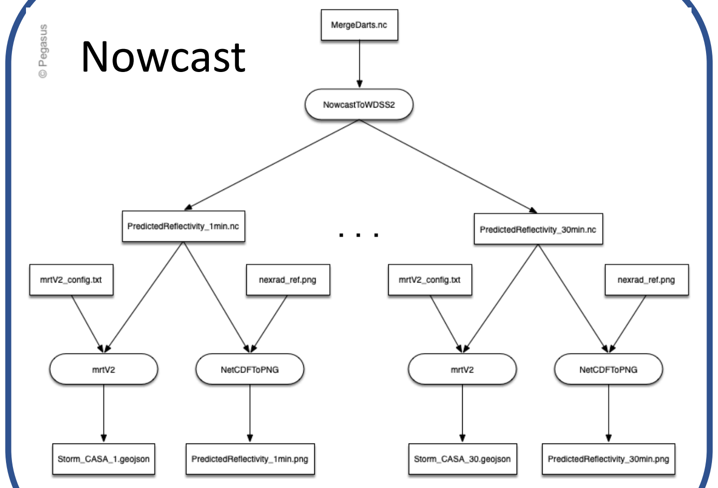
Tracking of rare events requires additional resources and dynamic resource provisioning capabilities

UMass Amherst DYNAMO renci RUTGERS USC Viterbi School of Engineering Information Sciences Institute

Dynamic resource provisioning

USC Viterbi
School of Engineering
Information Sciences Institute

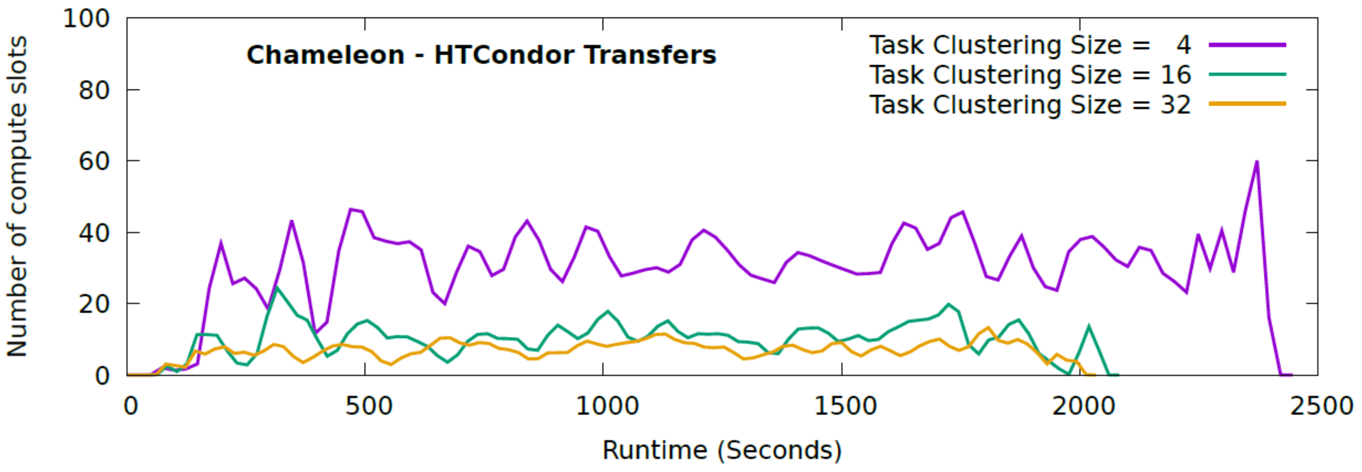


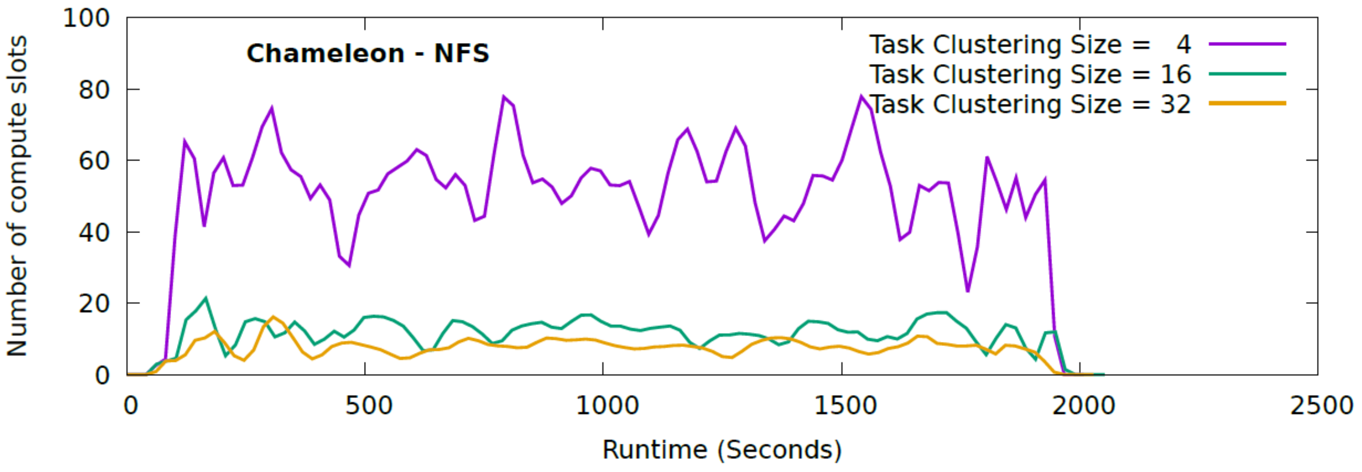


- Compute and storage resources on both ExoGENI and Chameleon clouds
- Dynamic resource provisioning on ExoGENI and Chameleon clouds
- High speed data movement via ExoGENI's dedicated layer-2 overlay networks
- Pegasus interacts with the Dynamo resource provisioners to acquire resources as needed

Evaluation – Required Resources by CASA Workflows

USC Viterbi
School of Engineering
Information Sciences Institute

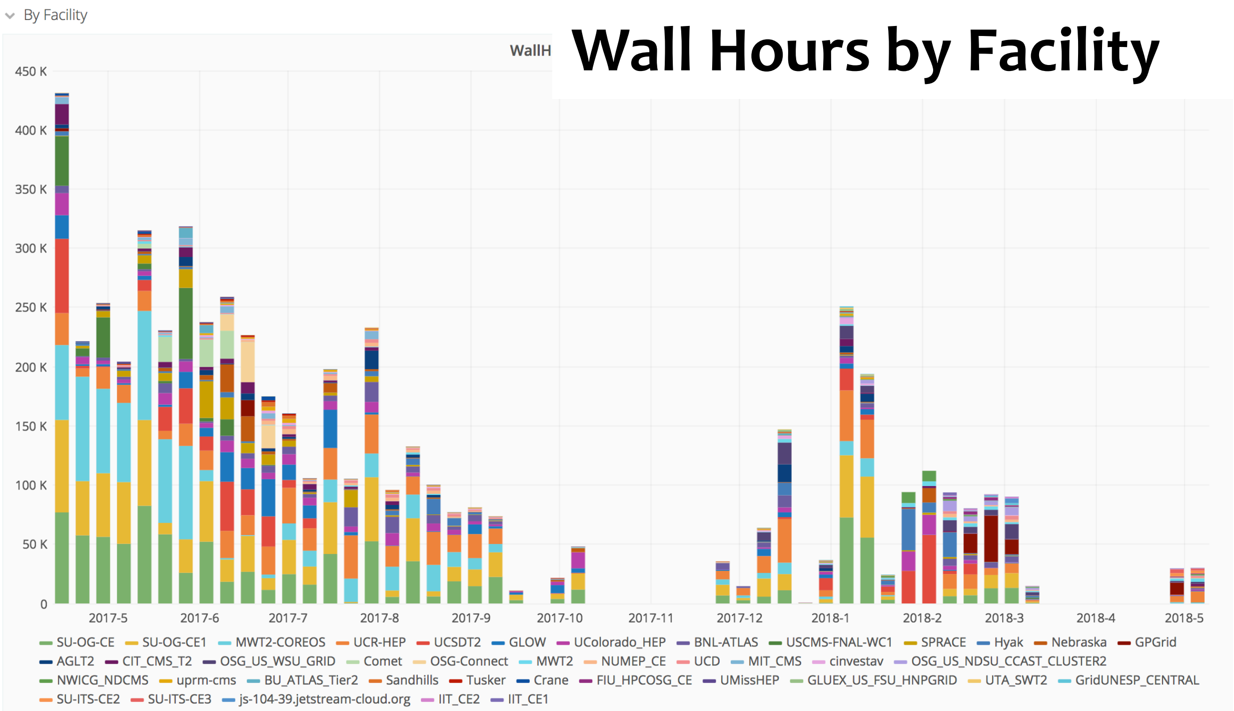




- Amount of resources required by compute intensive workflows like Nowcast
- Number of active compute slots for Nowcast
- Chameleon, with HTCondor transfers vs. using NFS
- Clustering of 4 tasks creates high demands (40-80 slots)
- Clustering of 16-32 decreases compute slot demand (<20 slots)

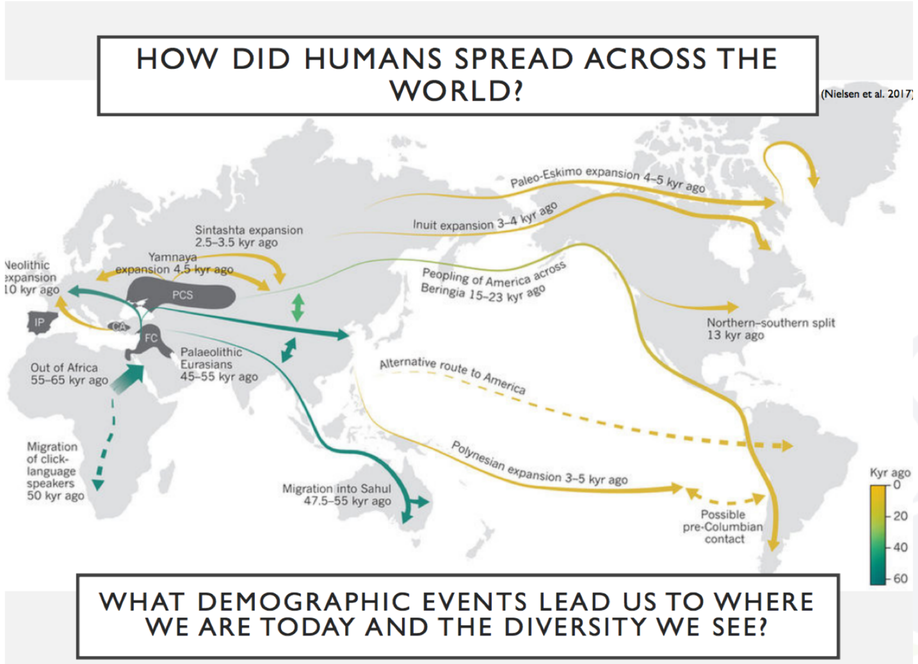
Arming Individual Scientists with Pegasus on OSG

Wall Hours by Facility



40 execution sites
12 million jobs across 342 workflows
~ 7.3 Million Wall Hours

Ariella Gladstein, Ph.D. Student
University of Arizona



Graph credit: Open Science Grid, Image credit: Gladstein

Looking ahead: Growing Demand for Automation

USC Viterbi
School of Engineering
Information Sciences Institute

HPC Systems

- Complex
- Heterogeneous
- Specialized data storage
- Increasingly faulty

Distributed Systems

- Software Defined capabilities
- Specialized data storage

Clouds

- New platform for science
- Very heterogenous
- Can be costly

Resource Management is Key

Constraints: time, budget

Faulty environment: detection and attribution

Heterogenous storage: memory, BB, FS, WAN

Workflow Ensembles can be challenging to manage

Tension between automation and transparency

Trust: How do you know that what we observe is real?

Inspect

Understand

Reproduce

Tension between automation and transparency

Increased need for
- automation
- autonomy

Role of ML

Current challenges increase

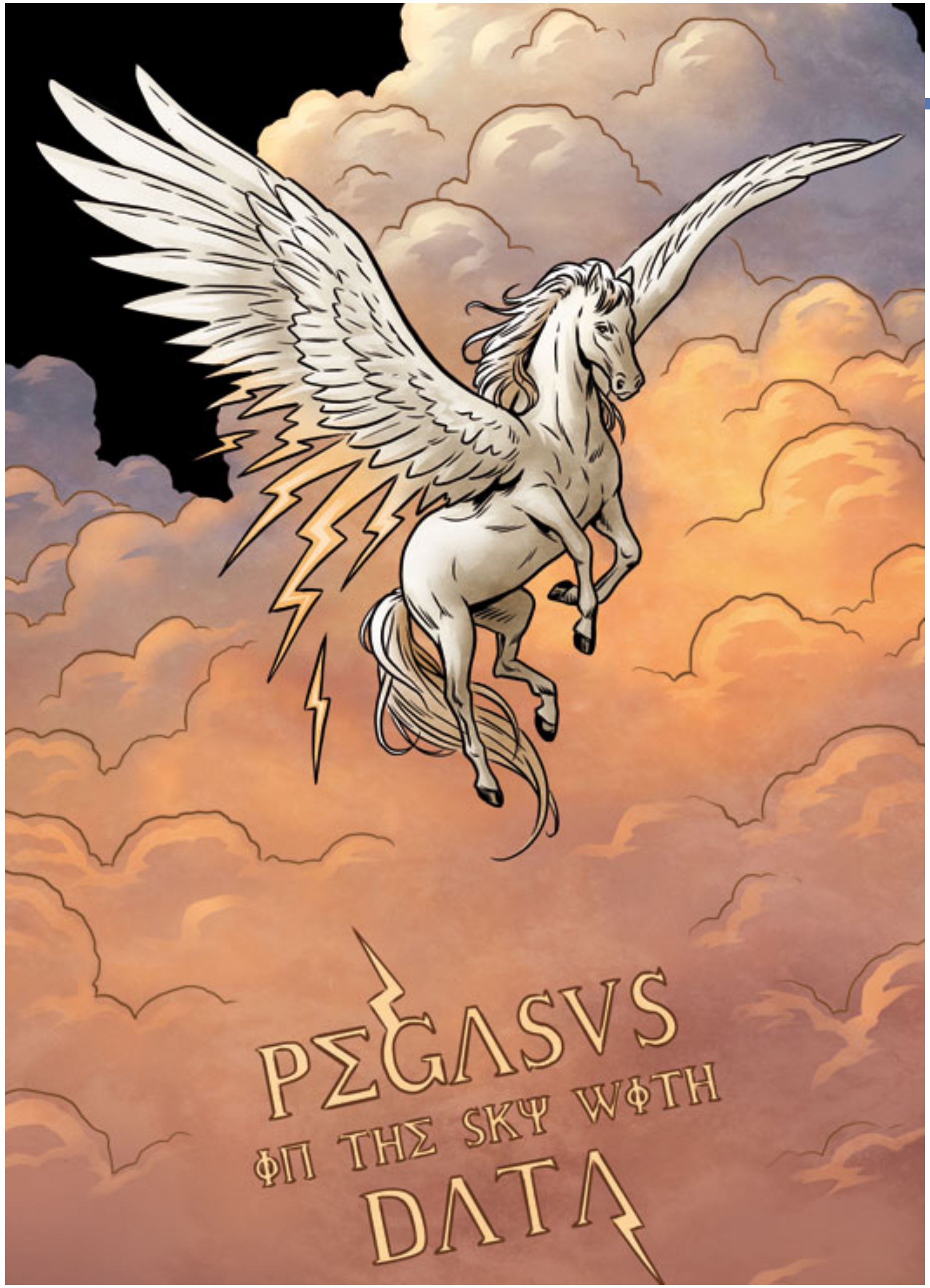
Trust: How do you know that what we observe is real?

Inspect

Understand

Reproduce

USC Viterbi Magazine, Spring 2018



USC Viterbi School of Engineering Information Sciences Institute

Conclusions

- Pegasus provides:
  - APIs for workflow composition in Python, R, Java, Perl, Jupyter Notebook
  - User-friendly monitoring and debugging tools
  - Automated data management
  - Workflow planning, and re-planning in case of failures
  - Optimization of workflow performance
  - Container management
  - Specialized workflow execution engines for HPC systems
  - Provenance tracking
  - Data integrity on data movement

Pegasus est. 2001

Automate, recover, and debug scientific computations.

We welcome the opportunity to work with new applications and enhance our solutions based on user's needs.

Pegasus Website

http://pegasus.isi.edu

Users Mailing List

pegasus-users@isi.edu

Support

pegasus-support@isi.edu