Scientific Workflows in a Heterogeneous World

Ewa Deelman, Ph.D.

USC Viterbi
School of Engineering
Information Sciences Institute

https://deelman.isi.edu

https://pegasus.isi.edu

Workflows Systems 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 performance 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
  - Different software, different performance profiles

  - Need to know system characteristics, scheduling interfaces

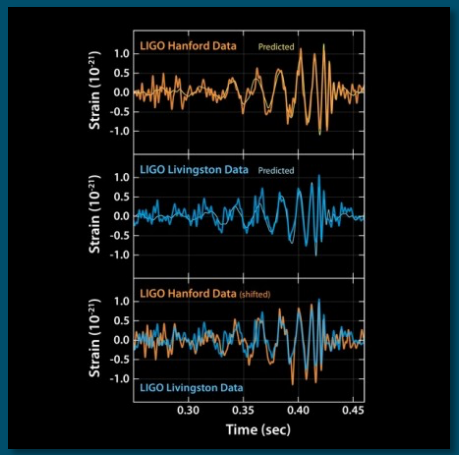
- Networks are highly heterogeneous but sometimes can be provisioned via SDN

Advanced LIGO PyCBC Workflow

One of the main workflows used to measure the statistical significance of data needed for gravitational wave detection

Contains 100's of thousands of jobs and accesses on order of terabytes of data

Uses data from multiple detectors



LIGO's resources: 11 large clusters at various institutions and affiliates
– Data is replicated at sites in the US and Europe
– Each cluster has Grid middleware and HTCondor installed
– GridFTP used for data transfers

Use our Pegasus software to automate the execution of tasks and data access

Pegasus

http://pegasus.isi.edu

Aerial View of the LIGO Livingston Laboratory
Image Credit: Caltech/MIT/LIGO Lab

LSC (LIGO Scientific Collaboration)

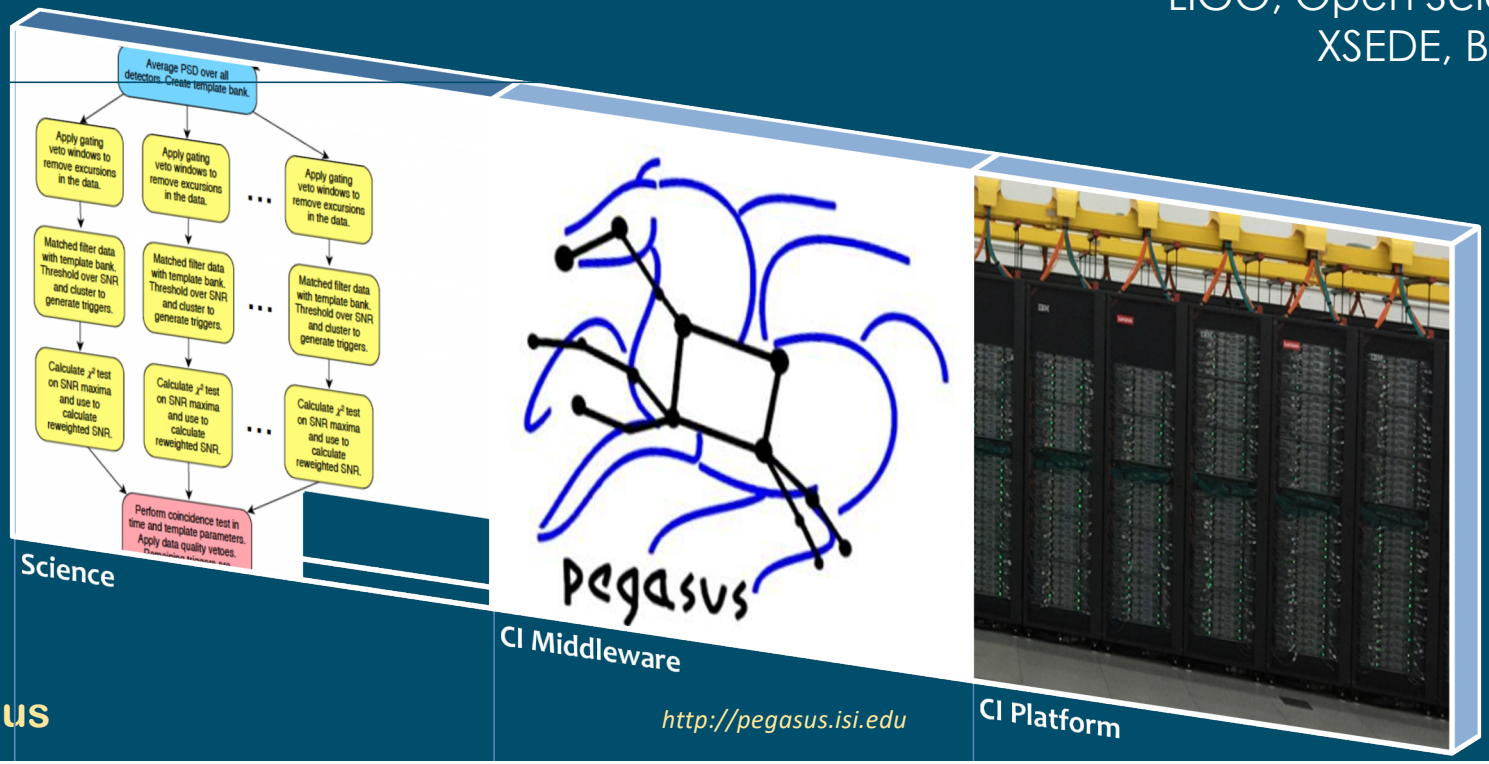
– Collaboration involved in research of the data coming out of the detectors
– 1000 scientists from universities in US and 14 other countries
– 250 students
– Responsible for developing analysis methodologies and detector technology.

First GW detection: ~ 21K workflows with ~ 107M tasks

Science workflow
measure the statistical significance of data needed for discovery

Automated by Pegasus
execution of tasks and data access

Distributed Power
LIGO, Open Science Grid, XSEDE, Blue Waters



2016

Workflow Management System (WMS) Functions

- Automate computational scientific methods that scientists rely on for their work
- Allow scientists to describe their workflows at a scientific level, in a resource-independent way
- Discover what resources (computation, data, software) are available
- Devise a plan and generate an executable workflow:
  - Select the appropriate resources based on an architecture, availability of software, performance, reliability, availability of cycles, storage,..
  - How to best adapt the workflow to the resources
  - Determined protocols to use to access the data, to schedule jobs
  - Determine data to save along the way
- Execute the plan
  - In a reliable and scalable way
  - Keep track of what data was accessed, generated and how: (provenance, reproducibility)
- Sometimes WMS provisions resources ahead of the execution (cloud, network)

Our Approach: Submit locally, Compute globally, workflow is abstract, helps support productivity

Work Definition

Pegasus Workflow Management System

Community Resource

Local HTCondor Pool

LIGO Data Archive

data

work

LIGO HTCondor Pool

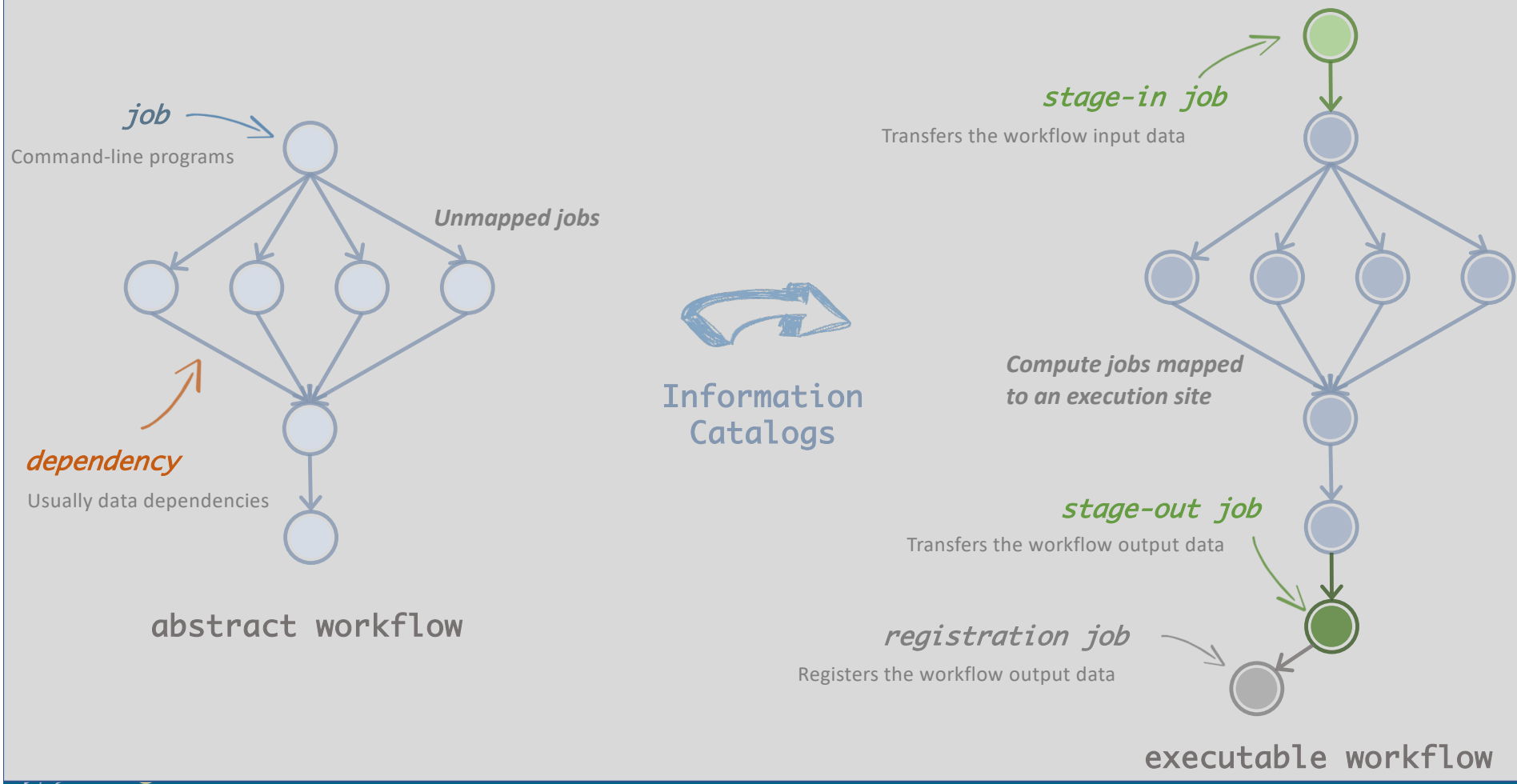
LIGO HTCondor Pool

LIGO Data Archive

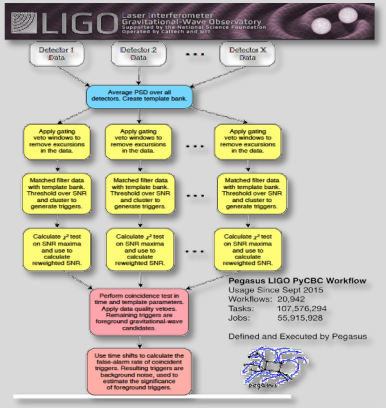
LIGO Data Archive

LIGO workflows are composed of High Throughput Computing (HTC) jobs

From the abstraction to execution



Easily add new resources because of the workflow abstraction



Work Definition

Pegasus Workflow Management System

Community Resource

Local HTCondor Pool

LIGO Data Archive

data

work

Blue Waters

XSEDE

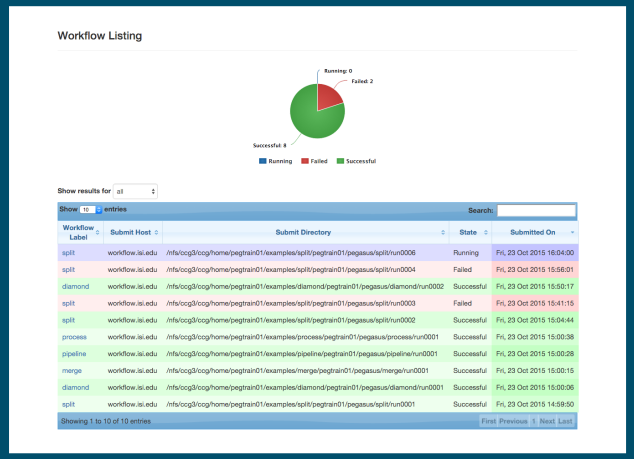
OSG

LIGO HTCondor Pool

LIGO HTCondor Pool

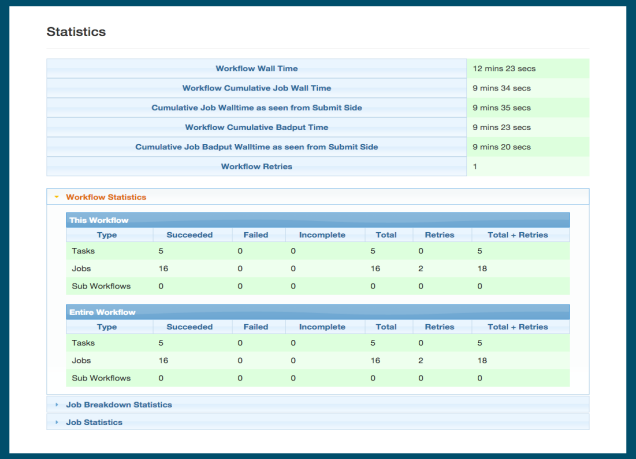
LIGO Data Archive

LIGO Data Archive

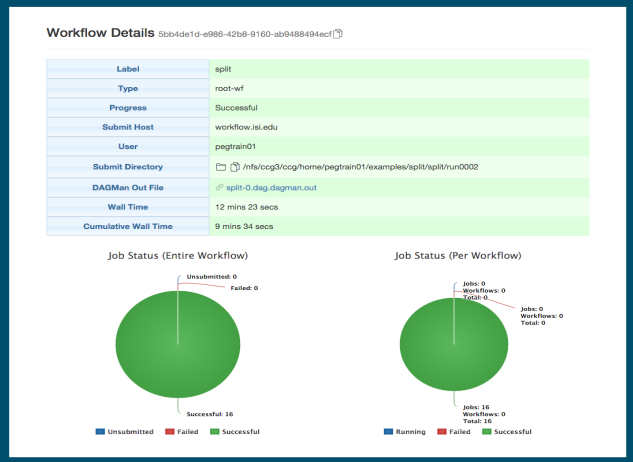


Pegasus dashboard

web interface for monitoring and debugging workflows



Real-time monitoring of workflow executions. It shows the status of the workflows and jobs, job characteristics, statistics and performance metrics. Provenance data is stored into a relational database.



Real-time Monitoring

Reporting

Debugging

Troubleshooting

RESTful API

Pegasus

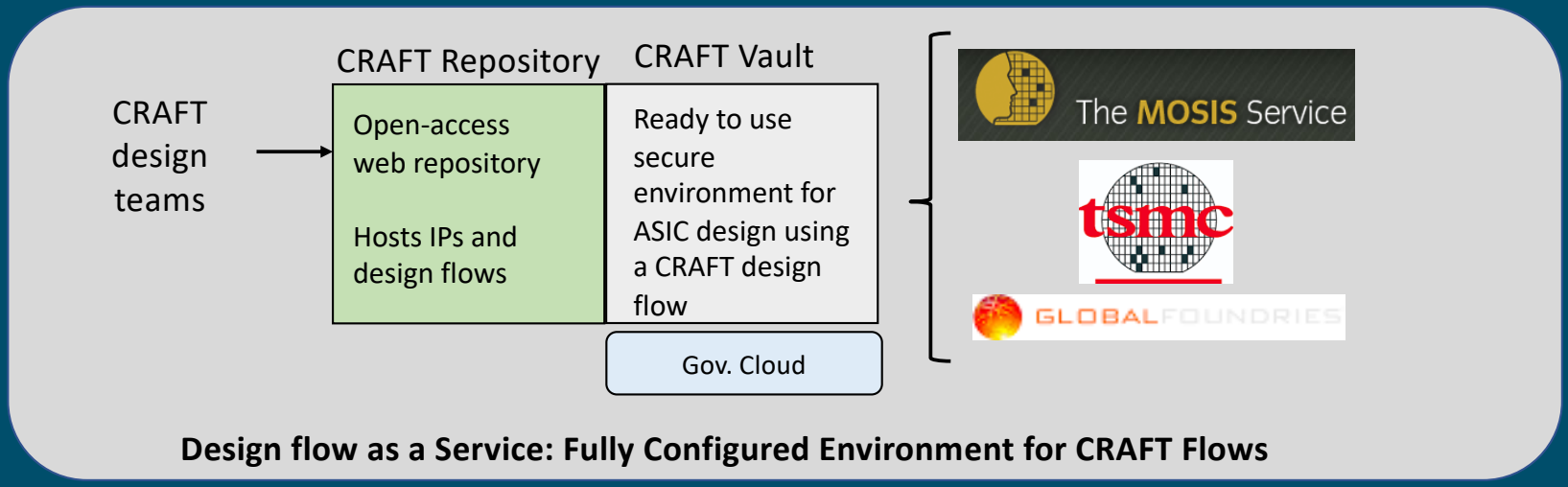
http://pegasus.isi.edu

Workflows in Circuit Realization at Faster Timescales (CRAFT)

RACE: Design Flow as a Service

- **Objective:** To enable **rapid**, **collaborative**, and **secure *development***, ***distribution***, and ***use*** of CRAFT design flows and IP for rapid design of complex ASICs/SoCs in advanced fabrication technologies
  - Design teams develop innovative design flows and IP to enable rapid design
  - We develop CRAFT Repository and Vault to enable design teams to rapidly learn and use of these design flows for ASIC/SoC design in advanced technologies

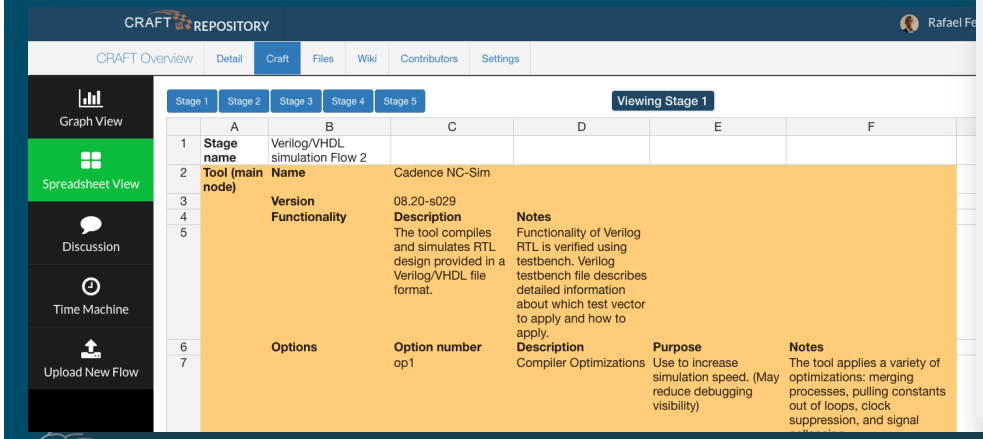
Collaboration with J. Nabrzyski, UND

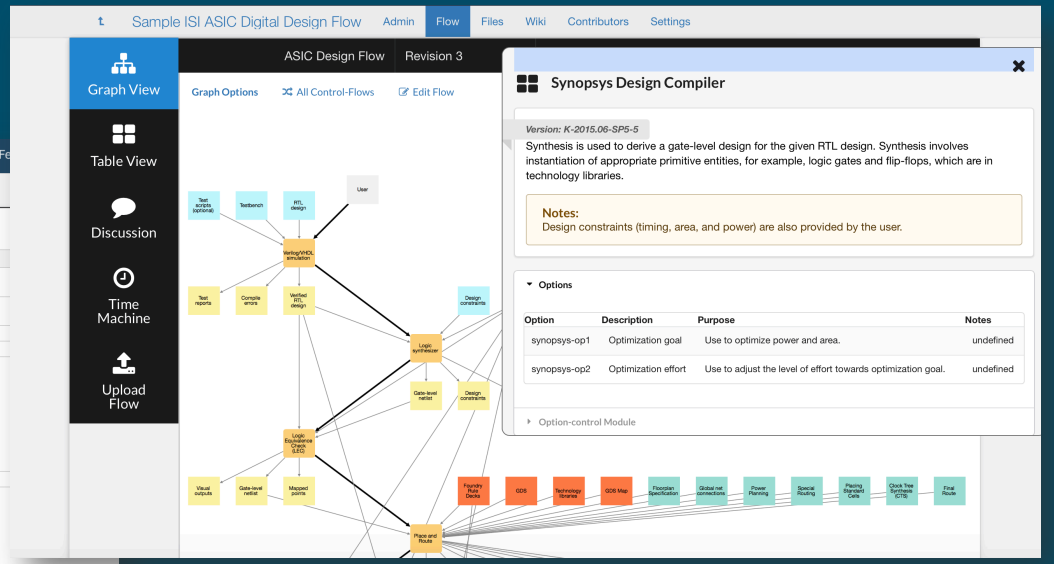


Design flow as a Service: Fully Configured Environment for CRAFT Flows

RACE: Design Flow as a Service

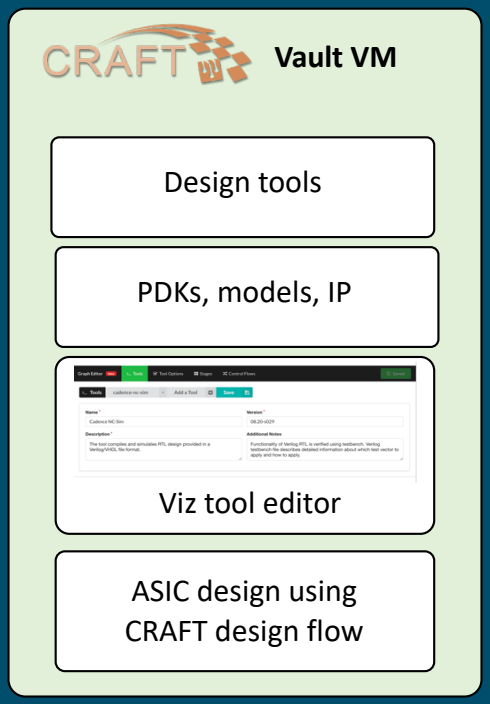
- A repository that:
  - Captures design flows with a great level of detail so that they can be understood and adopted by others
  - Catalogs IP used in the flows and allows for IP discovery and reuse
  - Provides data privacy and security
  - Allows collaboration between teams of designers; promotes sharing of ideas, experiences, and design flows





Vault for Design Flows, Virtual Environment

- Secure environment for collaboration and design
- Information and tools to follow and/or modify the design flow
- Allows for **mix of tools from multiple vendors**
- Leverages and augments repository information

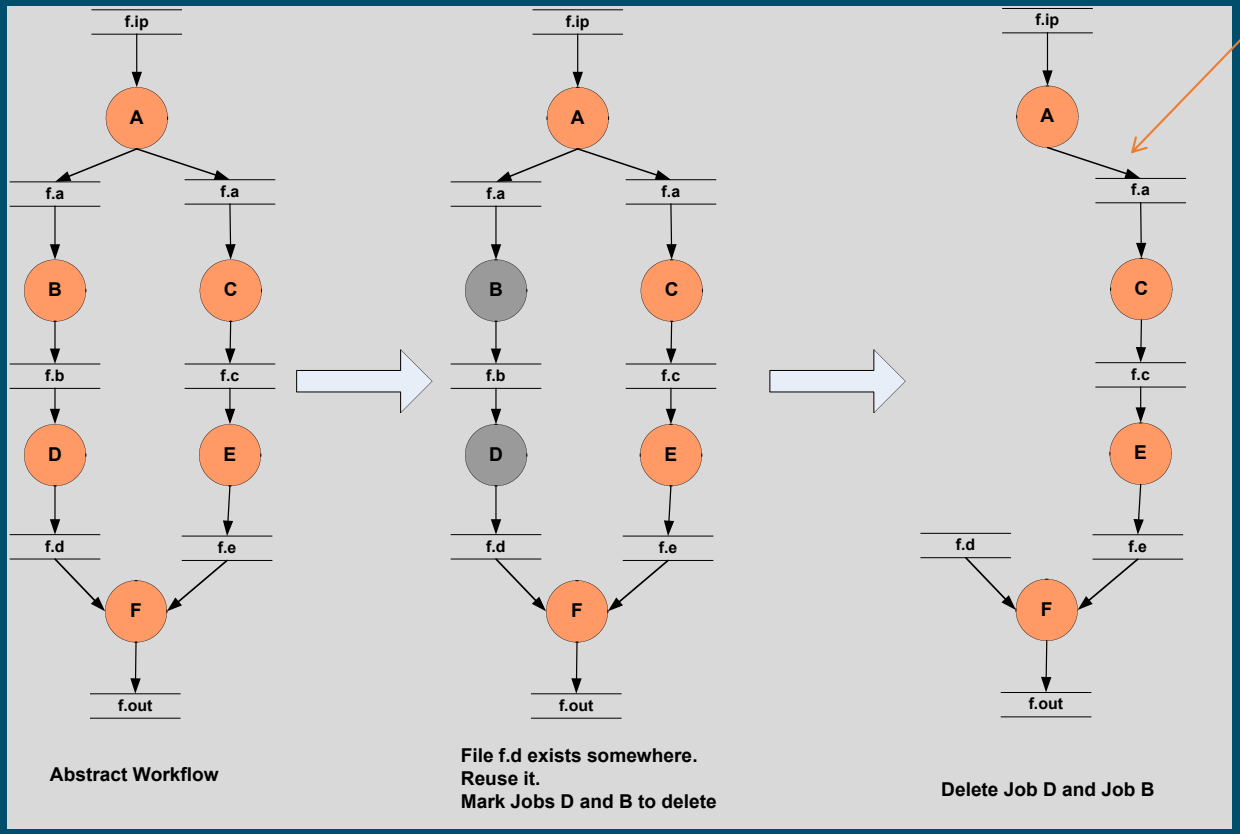


- Automated flow deployment
- An interactive design flow editor
- Provenance tracking of the entire design process for an ASIC/SoC, including exact versions of tools/scripts, PDKs, cell libraries, and IPs used
- *Adding human-in-the-loop to Pegasus*

If successful, we will make Extreme Heterogeneity worse!

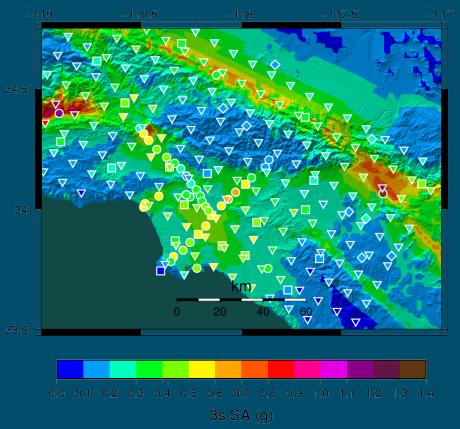
Data Re-use and Resilience

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



- *workflow reduction*
- *data reuse*
- *workflow-level checkpointing*

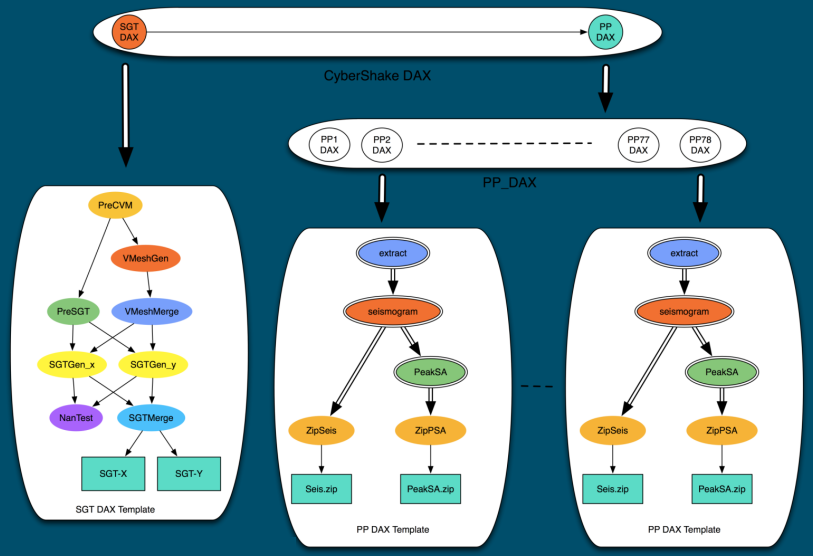
Southern California Earthquake Center
Mix of HPC and HTC codes



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)



Workflow Ensembles: 239 Workflows

- Each site in the input map corresponds to one workflow
- Each workflow has:

  820,000 tasks

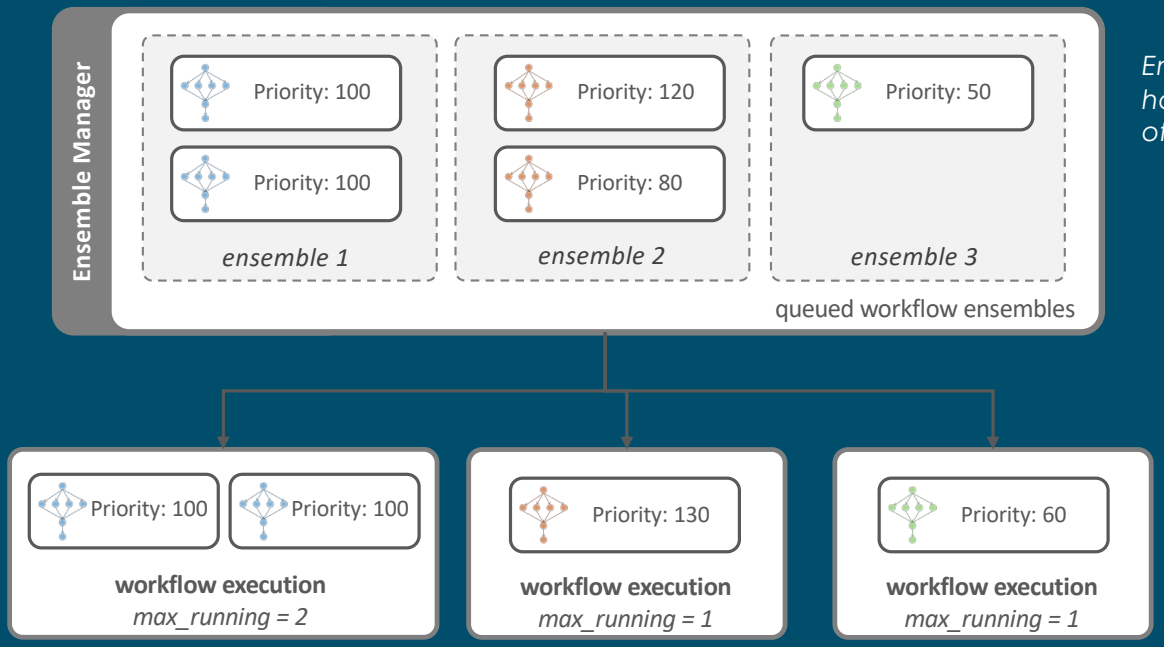
Runs on HPC (Blue Waters, Titan)

Runs on HTC or HPC

Pegasus

http

Running Workflow Ensembles



*Ensemble Manager can handle **multiple ensembles** of workflows independently*

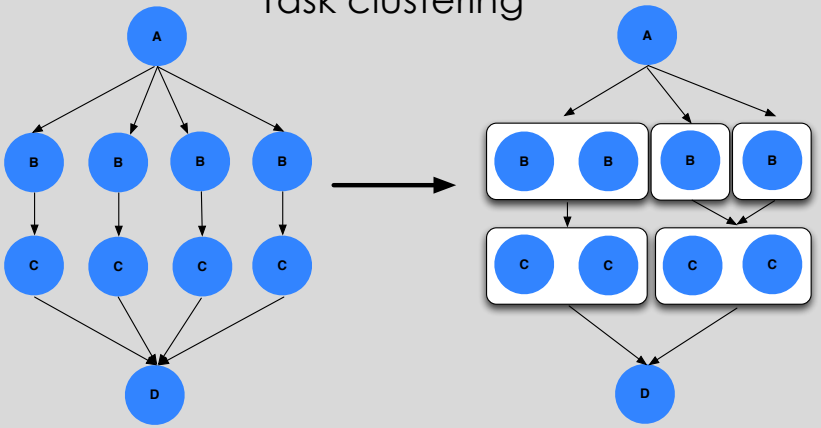
***concurrent** planning and execution of multiple workflows*

- Workflows within an ensemble may have different priorities that *can also be changed at runtime*
- May limit the number of concurrent planned and running workflows
- Ensembles can be paused, resumed, removed, re-planned, and re-executed
- A Debugging mechanism is also provided to investigate failures in workflow runs

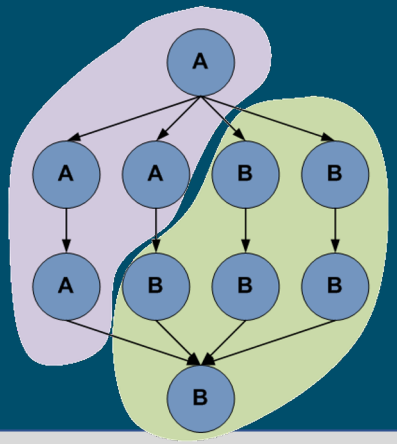
Prototype feature: *support for **event triggers** that can trigger addition of new workflows to an existing ensemble*

Solutions to running HTC on HPC:

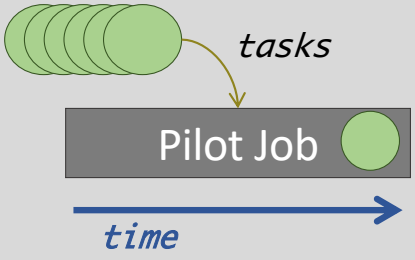
Task clustering

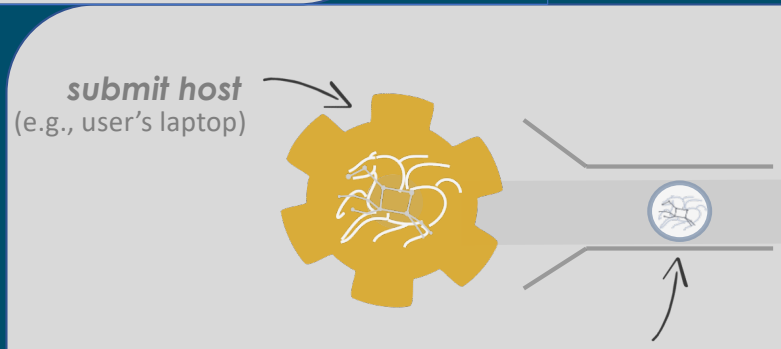


Partition the workflow into sub-workflows and send them for execution to the target system



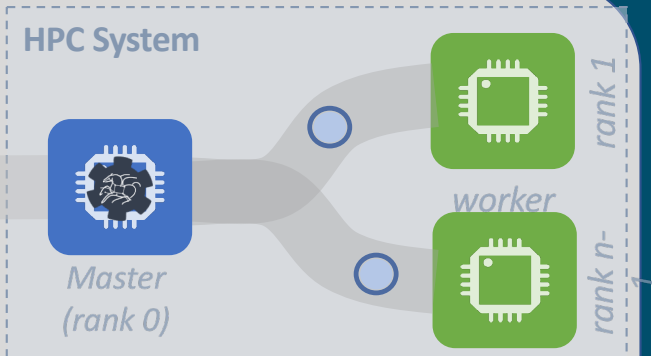
Use "pilot" jobs to dynamically provision a number of resources at a time





workflow wrapped as an MPI job

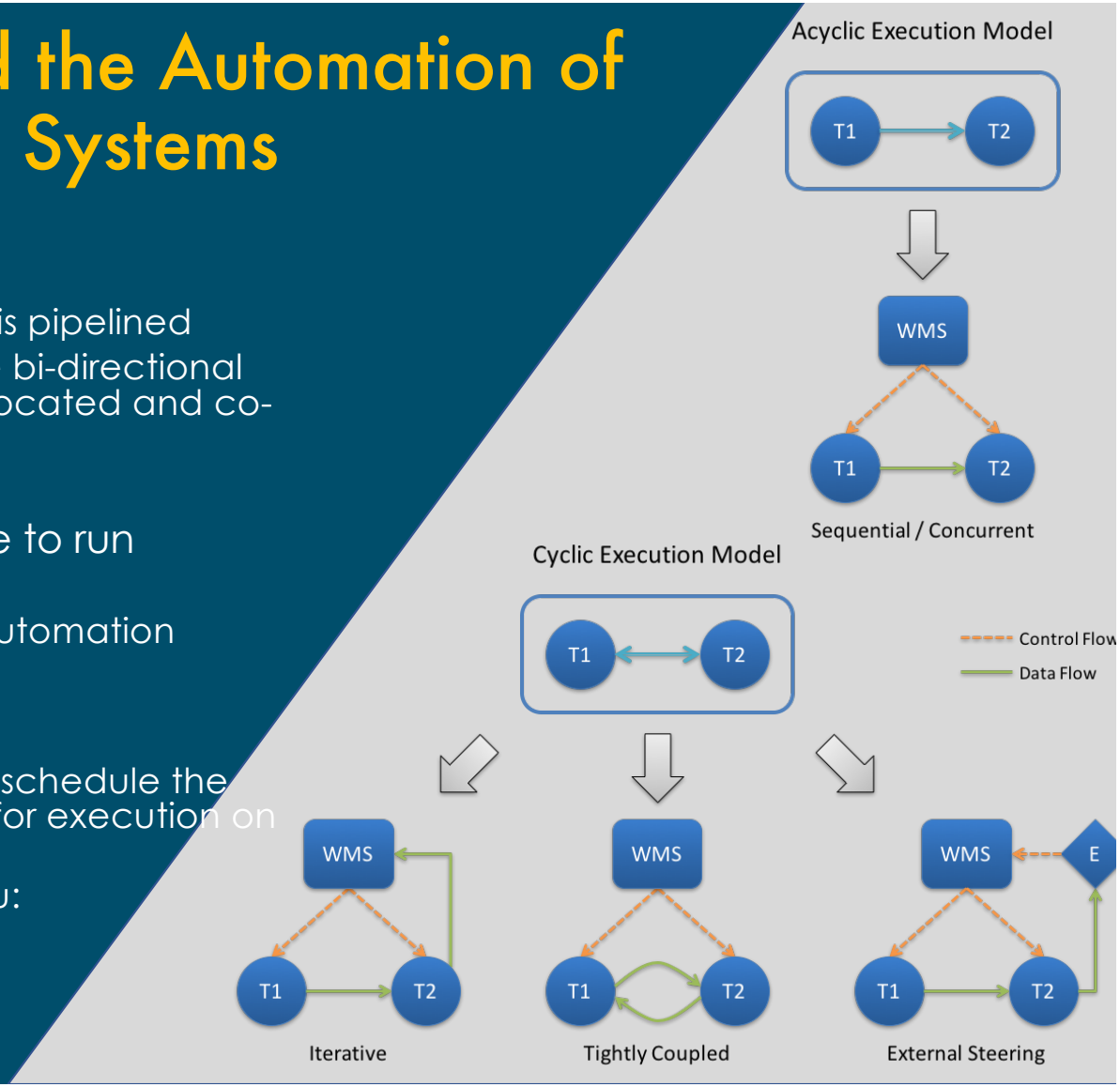
Allows sub-graphs of a Pegasus workflow to be submitted as monolithic jobs to remote resources



PMC (Pegasus MPI Cluster), a specialized workflow engine

New Applications Need the Automation of Workflow Management Systems

- Todays workflows
  - independent high-throughput tasks
  - loosely couple tasks, where data flow is pipelined
  - tightly coupled applications that have bi-directional communications and need to be co-located and co-scheduled (Michela's apps)
  - a mix of all
- Users don't want to worry about where to run
  - need results in a timely manner
  - need ease of design, ease use, and automation
- Workflow Management Systems
  - can cross system boundaries
  - can select the appropriate resources, schedule the needed data movement, send tasks for execution on the target resources
  - systems that couple applications in-situ: Decaf, DataSpaces, ADIOS



Pegasus

Challenges of Workflow Management on Extreme Heterogeneous Systems

- Applications:
  - Tasks can have different resource needs
  - The needs may not be fully known ahead of the execution
  - There may be a number of inter-related workflows running at the same time (workflow ensembles)
- HPC Systems:
  - Resource discovery is based on static information about general availability
  - Limited dynamic information from the scheduler
  - Limited resource provisioning capabilities, but there are some good examples
    - Can reserve burst buffers as part of the workflow
    - Can share burst buffers across workflows to support data reuse
- Main issue: who controls the resources?
  - Traditional schedulers do fine-grained resource assignments, may not be scalable or desirable for extreme, highly heterogeneous systems
  - In an alternative view WMS could request a set of resources from the scheduler and manage them on behalf of the workflow or workflow ensemble
    - Doing it indirectly today with pilot jobs

Resource control

- WMS knows the current and potentially future workload needs at a coarse level so it can potentially make better decisions
- Resource discovery for WMS scheduling:
  - Number of available cores, CPUs, GPUs, memory
  - Available of various architectural features, such as burst buffers and their capacity, interfaces
  - Discovery of data location (if you pre-staged data into a bb or node-local disk for example)
- Resource provisioning:
  - Not just focused on compute resources
  - Could be asking for:
    - Compute resources "close to data sets"
    - The same compute resources as job *j-1* had
- Questions?
  - What information to expose about the target system(s)?
  - How does this information need to be aggregated/what's the right level of abstraction?

Job scheduling

- Need task performance models that take into account the heterogeneous system features
  - Mix of analytical models, simulation, machine learning
- Need models that take into account data movement within the storage hierarchy: memory, burst buffers, file system, via LAN, via WAN
- Focus on data-aware scheduling for an ensemble, workflow, job, within a job (Torsten)
- Need more functionality from schedulers to support changing application demands over time (de-allocate/allocate)

Online Application and System Monitoring

- Besides information needed for making resource provisioning and scheduling decisions, need information during execution to:
  - Capture performance and faults
  - Potentially discover new data sets
  - Any other information that can help WMS make decisions about releasing/acquiring resources
    - New resources available?
- WMS needs to be self-aware, assure its own execution does not affect the workflows it managed
  - Assuming the WMS or a specialized workflow engine is running in-situ along side the simulation and analytics
- Questions?
  - Today's information systems– e.g. Darshan collect a lot of information, how can it be synthesized or analyzed to provide near real-time, actionable information?
  - How can ML techniques help?
  - How do you divide the responsibility for the health of the application and system between the scheduler and WMS?

Fault Tolerance

- Performance models and/or ML can be used for anomaly detection
- Need algorithms to quickly detect anomalies
- WMS could adapt, suspend, abort workflow, or trigger an alert
- Need to co-manage the execution of applications, anomaly analysis, WMS
- Need more sophistication from WMS to make scheduling and adaptation decisions
- Need better fault tolerance in HPC/WMS environments
  - 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/NVM/disk (time to solution and energy consumption)

Other things we learned in distributed WMS:

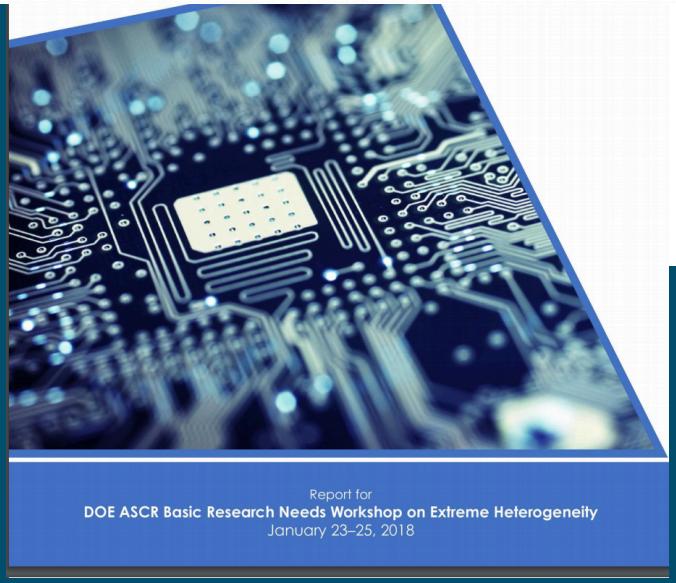
- Provenance Capture and Reproducibility: WMS capture provenance information about the creation, planning, and execution
  - Up to now, the approach has been to save everything – problem
  - Provenance capture may need to adapt to the behavior of the application (coarse and fine levels of details, compression)
  - May want to automatically re-run parts of the computation and re-produce the results and a more detailed provenance trail on demand
  - Systems need to be made reproducibility aware: provide enough introspection to reason about the validity and reproducibility of the results
  - Still need provenance exploration tools

Conclusions

- Keep the separation between the workflow description and the executable workflow
- Take into account heterogeneous workflow applications and workflow ensembles
- There is no single WMS that easily traverses HTC and HPC boundaries although scientific applications do
  - Need new workflow technologies that are more versatile
  - Explore various data communication modalities within workflows, taking full advantage of system architectural features. (memory, burst buffer, file system)
  - Make sure to include data flow from DTNs and other remote storage
  - Develop better resource provisioning and job scheduling capabilities
- There are opportunities to define new schedulers and new scheduler/WMS interactions
  - Need more information and control flow
  - Need for abstraction for information flow
  - Need to combine resource provisioning and job scheduling
- Layering is important: relationship to runtime systems

Extreme Heterogeneity 2018

PRODUCTIVE COMPUTATIONAL SCIENCE IN THE ERA OF EXTREME HETEROGENEITY



Workshop Organizers

Jeffrey S. Vetter	Oak Ridge National Laboratory
Lucy Nowell	DOE Office of Advanced Scientific Computing Resear
Katie Antypas	Lawrence Berkeley National Laboratory
Ron Brightwell	Sandia National Laboratories
David Donofrio	Lawrence Berkeley National Laboratory
Maya Gokhale	Lawrence Livermore National Laboratory
Travis Humble	Oak Ridge National Laboratory
Pat McCormick	Los Alamos National Laboratory
Rob Ross	Argonne National Laboratory
Catherine Schuman	Oak Ridge National Laboratory
John Shalf	Lawrence Berkeley National Laboratory
Brian Van Essen	Lawrence Livermore National Laboratory
Shinjae Yoo	Brookhaven National Laboratory

- "Understanding the trade-offs between productivity, portability, and performance metrics for scientific workflows
- Raising the level of abstraction in workflow descriptions to hide implementations and platforms "

Pegasus Team at the USC Information Sciences Institute

2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019

Loïc Pottier

Back Row: Tu Mai Anh Do, Karan Vahi, Mats Rynge, George Papadimitriou, Rafael Ferreira da Silva
Front Row: Rajiv Mayani, Ryan Mitchell, Ragini Chugh, Ewa Deelman, Mukund Murrali
Missing: Loïc Pottier

Pegasus

http://pegasus.isi.edu

Ewa Deelman, deelman@isi.edu