SSI: Distributed Workflow Management Research and Software in Support of Science

USC Viterbi
School of Engineering

Ewa Deelman, Karan Vahi, Mats Rynge, Rajiv Mayani, George Papadimitriou, Aditi Jain - *USC Information Sciences Institute*
Miron Livny, Cole Bollig- **University of Wisconsin, Madison**

Pegasus WMS

Pegasus is a system for mapping and executing abstract application workflows over a range of execution environments.
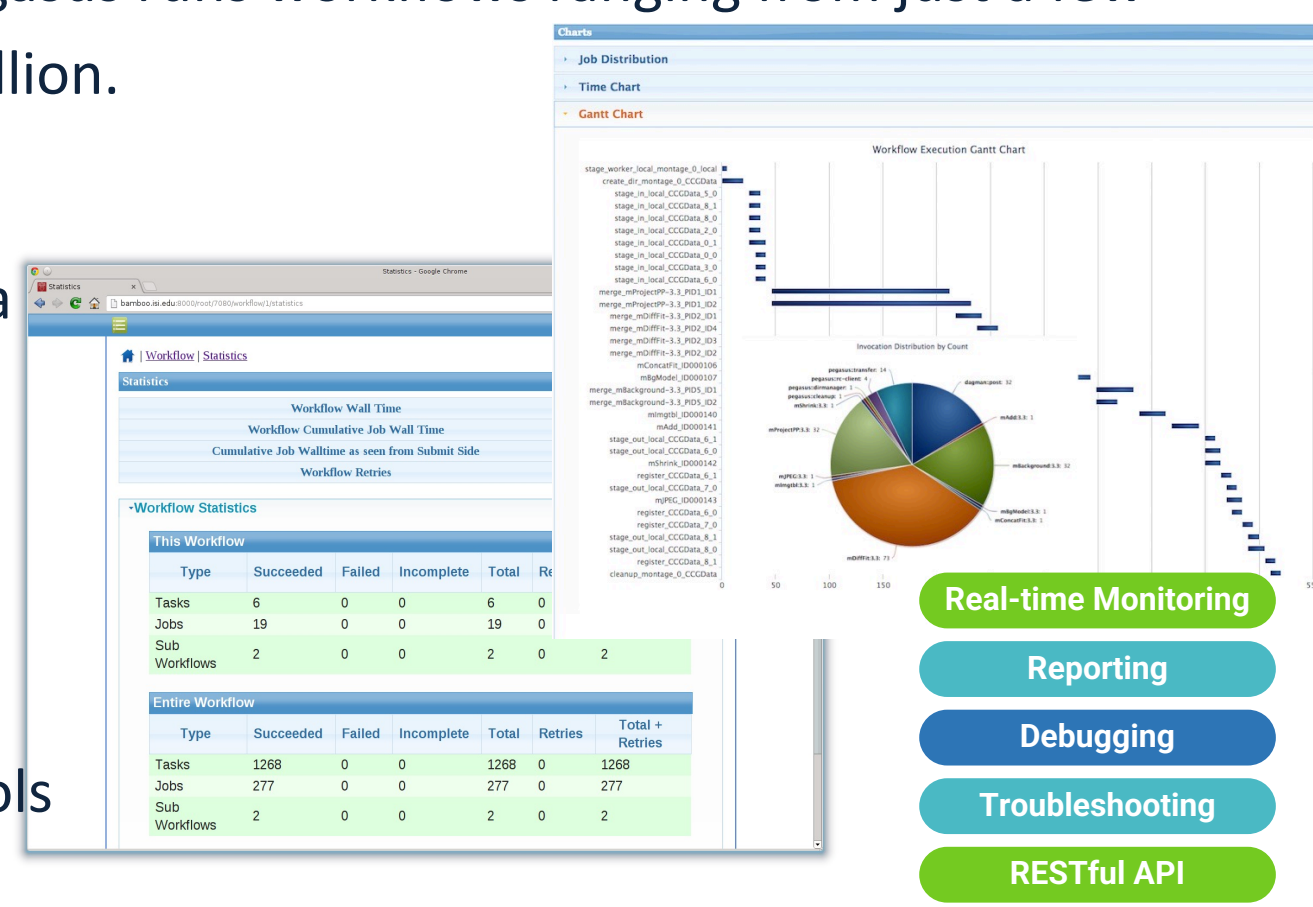
The same abstract workflow can, at different times, be mapped different execution environments such as XSEDE, OSG, commercial and academic clouds, campus grids, and clusters.

Pegasus can easily scale both the size of the workflow, and the resources that the workflow is distributed over. Pegasus runs workflows ranging from just a few computational tasks up to   1 million.

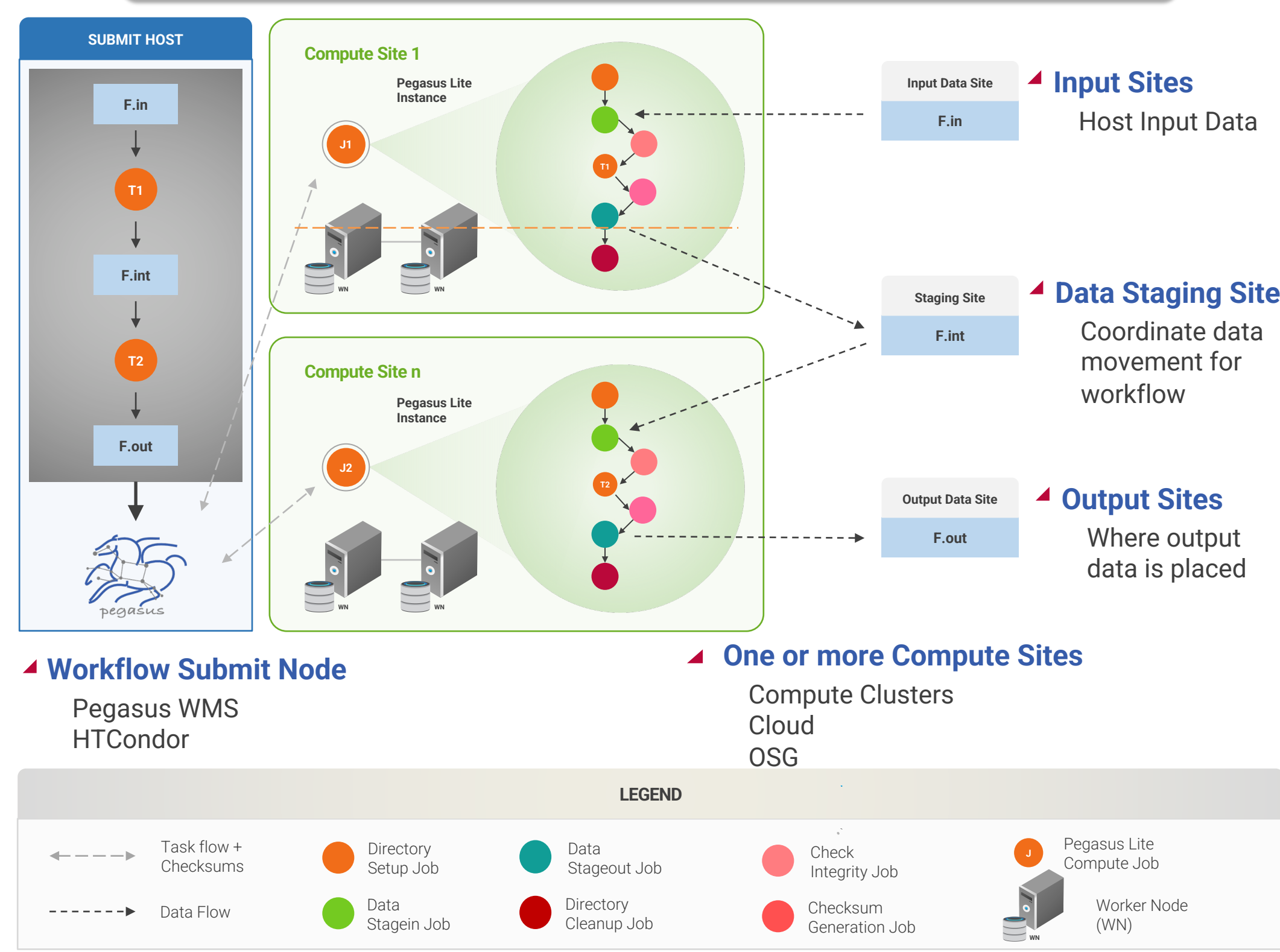End to End automatic checksumming of workflow data to ensure data integrity.

Stores static and runtime metadata associated with workflow, files and tasks. Accessible via command line tools and web-based dashboard.

- Real-time Monitoring
- Reporting
- Debugging
- Troubleshooting
- RESTful API

Pegasus Deployment

Workflow Submit Node
Pegasus WMS
HTCondor

One or more Compute Sites
Compute Clusters
Cloud
OSG

LEGEND

Task flow + Checksums	Directory Setup Job	Data Stageout Job
Data Flow	Data Stagein Job	Directory Cleanup Job
		Check Integrity Job
		Checksum Generation Job
		Pegasus Lite Compute Job
		Worker Node (WN)

Input Sites — Host Input Data

Data Staging Site — Coordinate data movement for workflow

Output Sites — Where output data is placed

Applications using Pegasus

Astronomy and Physics
- Pegasus powered workflows help **detect gravitational waves**.
- **DECam** workflows to **detect** optical counterparts to LIGO detections.
- XENONnT workflows for searching Dark Matter.
- Event Horizon Telescope for creating images of Black Holes.
- Galactic Plane workflow generates mosaics for astronomy surveys.
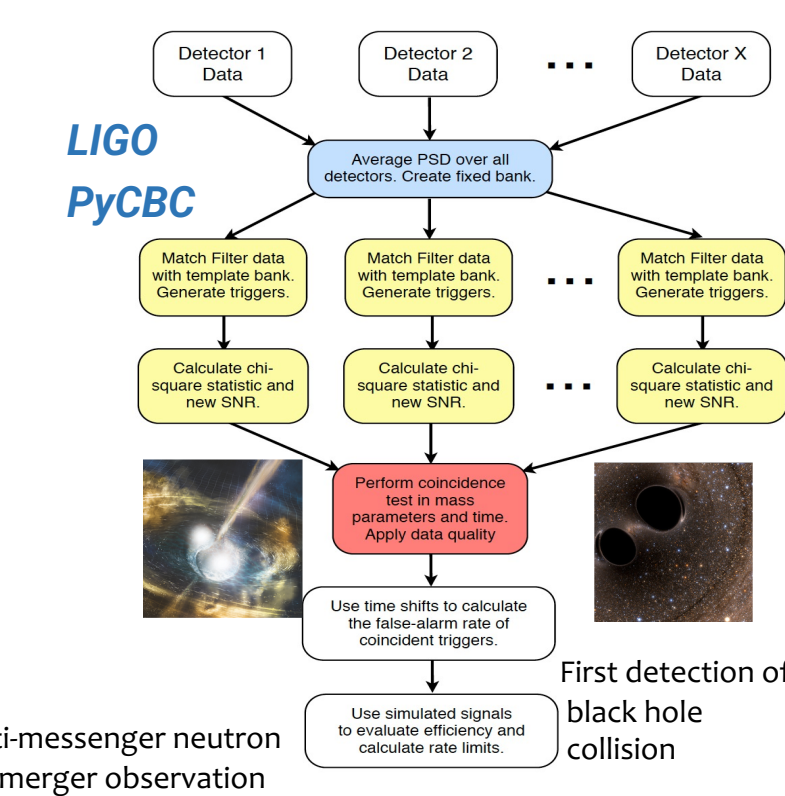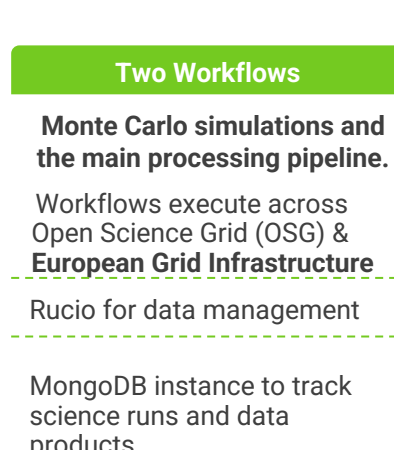
Seismology
- CyberShake workflows for seismic hazard analysis of LA basin.
- Broadband workflows for accurate predictions of ground motions.

Ecology
- Integrated Assessment Models to project impact of policy scenarios On socio environmental systems
- Predicting Flash Floods in Dallas FortWorth Metroplex
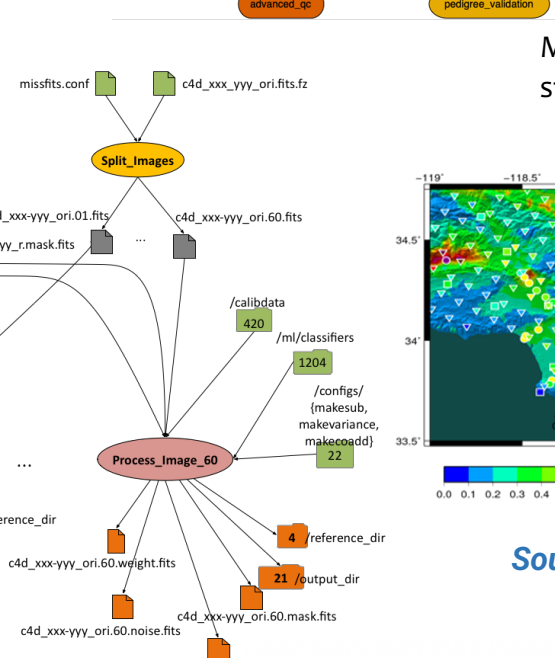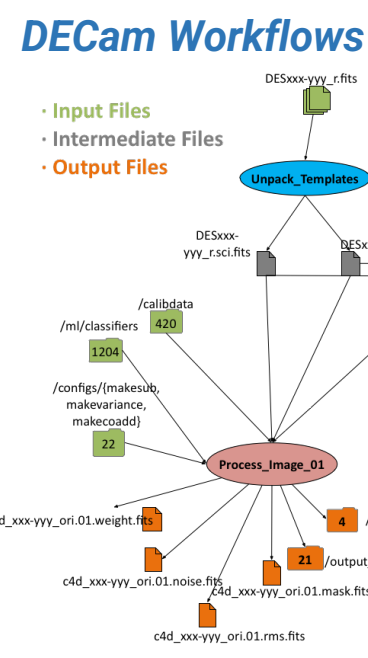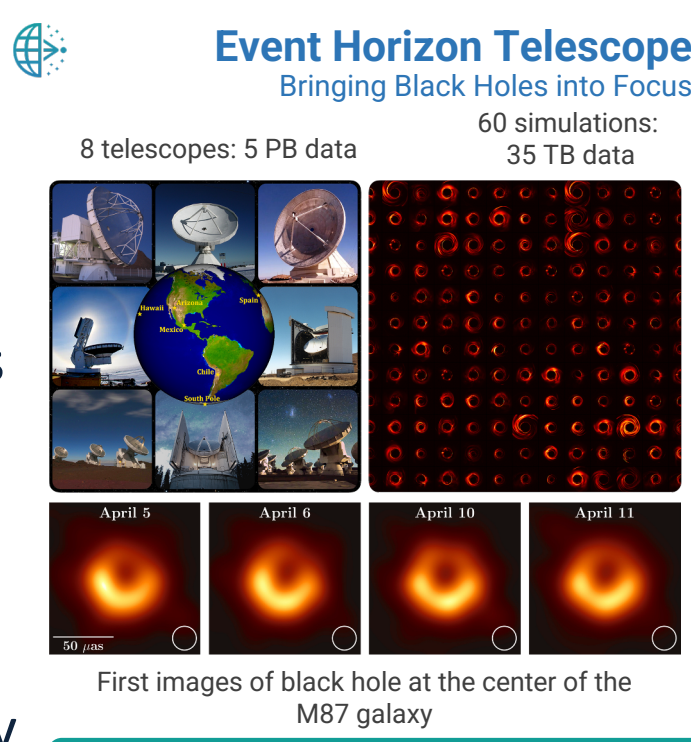
Microscopy
- Investigation of Strong Nuclear Force using gamma ray spectroscopy
- Cryo-EM Electroscopy for 3D reconstruction of biological samples

Bioinformatics
- Quality control workflows for data submissions to NRGR repository
- Imputation workflows on PAGE data.
- Workflows for Genome and Transcriptome free analysis of RSEQ.
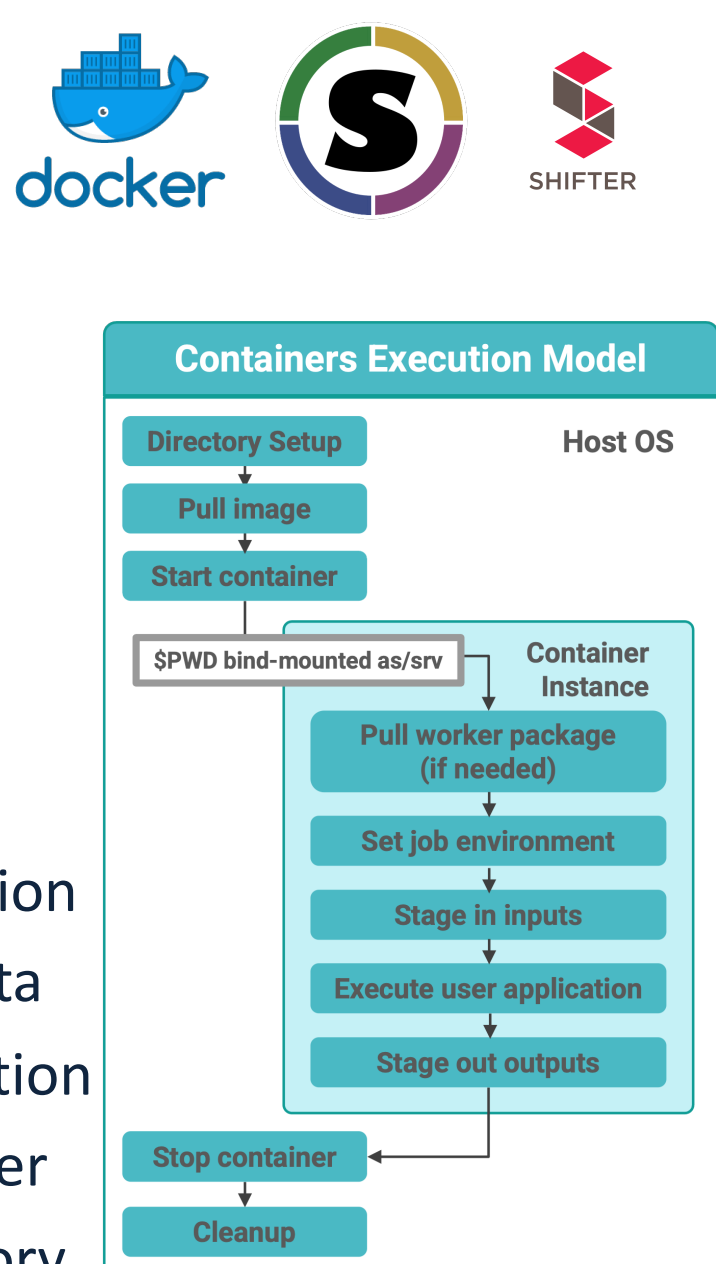- Brain span workflows help study gene expression in the brain.

Others http://pegasus.isi.edu/applications

XENONnT

Two Workflows
Monte Carlo simulations and the main processing pipeline.
Workflows execute across Open Science Grid (OSG) & European Grid Infrastructure
Rucio for data management
MongoDB instance to track science runs and data products.

NRGR QC

LIGO PyCBC
First detection of black hole collision

Multi-messenger neutron star merger observation

Event Horizon Telescope
Bringing Black Holes into Focus
8 telescopes: 5 PB data
60 simulations: 35 TB data
April 5   April 6   April 10   April 11
First images of black hole at the center of the M87 galaxy
Improve constraints on Einstein's theory of general relativity by 500x

DECam Workflows
- Input Files
- Intermediate Files
- Output Files

Mix of MPI and single-core jobs, mix of CPU, GPU codes. Large data sets (10s of TBs), ~300 workflows with 420,000 tasks each
Supported since 2005: changing CI, x-platform execution

Southern California Earthquake Center Cybershake

Vermont EPSCoR Integrated Assessment Model

Integrated Assessment Models (IAMs)
IAMs project the impact of policy scenarios on socio-environmental systems (SESs) and thus, provide policy makers with data to make informed policy decisions.

Heterogeneous Execution Platforms – University of Vermont (UVM) and UCAR's Cheyenne

Parallelism achieved over different policy scenarios (max of 176 so far) to enable bidirectional asynchronous feedback between models

176 Scenario, 60 Year IAM Run: ~ 10 days wall time, ~ 25,000 jobs, ~ 3.1TB data generated

CryoEM Processing

Computational Processing on GPU Cluster
Pegasus WMS

Containers

- Application containers provides a solution to package software with complex dependencies to be used during workflow execution
- Users have the option of either using a **different container for each executable** or **same container for all executables.**

Container Execution Model
- Container image is put in the job directory along with input data.
- Loads the container if required on the node (applicable for Docker)
- Run a script in the container that sets up Pegasus in the container and job environment
- Stage-in job input data
- Launches user application
- Ship out the output data generated by the application
- Shut down the container
- Cleanup the job directory

Containers Execution Model
Host OS
- Directory Setup
- Pull image
- Start container
- SPWD bind-mounted as/srv

Container Instance
- Pull worker package (if required)
- Set job environment
- Stage in inputs
- Execute user application
- Stage out outputs

- Stop container
- Cleanup

Pegasus 5.0

- New and fresh Python3 API to compose, submit and monitor workflows, and configure catalogs
- Python3 Support
  All Pegasus tools are Python3 compliant Python PIP packages for workflow composition and monitoring
- Zero Configuration to submit to local HTCondor Pool
- Moved to YAML representations
- Reworked Documentation and Tutorial
  https://pegasus.isi.edu/documentation/

Data Management Improvements

- Output Replica Catalog to track outputs
- Improved support for hierarchical workflows

```python
#!/usr/bin/env python3
import logging
import sys

from Pegasus.api import *

# logs to be sent to stdout
logging.basicConfig(level=logging.DEBUG, stream=sys.stdout)

# --- Transformations ------------------
echo = Transformation(
    "echo",
    pfn="/bin/echo",
    site="condorpool"
)

tc = TransformationCatalog()\
    .add_transformations(echo)

# --- Workflow ------------------
Workflow("hello-world", infer_dependencies=True)\
    .add_jobs(
        Job(echo)
            .add_args("Hello World")
            .set_stdout("hello.out")
    ).add_transformation_catalog(tc)\
    .plan(submit=True)\
    .wait()
```

Software Availability

Release Schedule
- Major Release every 9 months. Minor releases every 4 months
- Continuous Integration Testing with Bamboo
- Issue Tracking via JIRA

Download Options
- Source Code publicly hosted on GitHub
- Binary packages for Linux and MAC
- YUM/APT repositories with RPM/DEB packages
- Nightly Developments builds also available

Documentation / Training Materials
- Tutorials – Jupyter Notebooks in Docker containers
- Support – Slack, Email lists and Online User Guide

Downloads & Usage Since 2013
- Workflows - 2,222,077
- Tasks   - 6,789,748,851
- Jobs   - 1,675,182,509
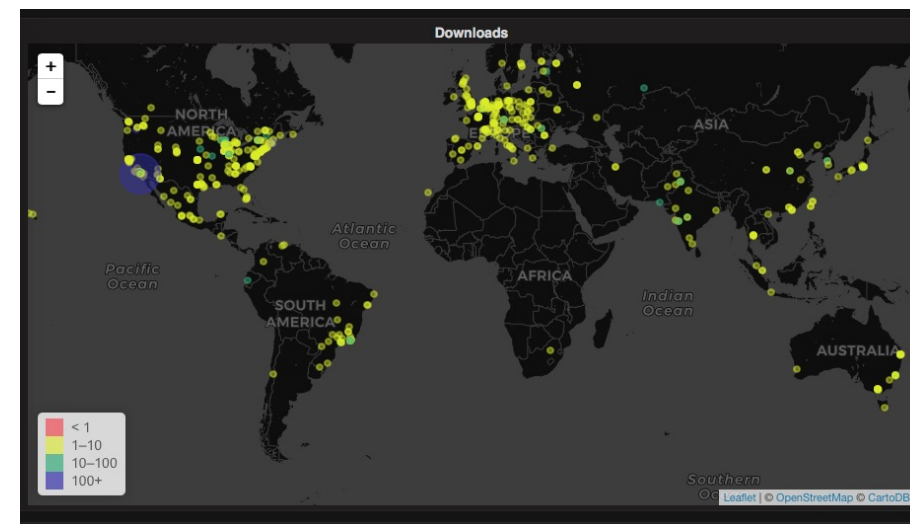
Year	Downloads
2012	62
2013	980
2014	874
2015	1,981
2016	5260
2017	14,474
2018	20,878
2019	23,685
2020	36,920
2021	47,665

HTCondor High Throughput Computing

http://pegasus.isi.edu

USC Viterbi
School of Engineering

THE UNIVERSITY
WISCONSIN
MADISON