Open Science Grid

Submit locally, run globally.

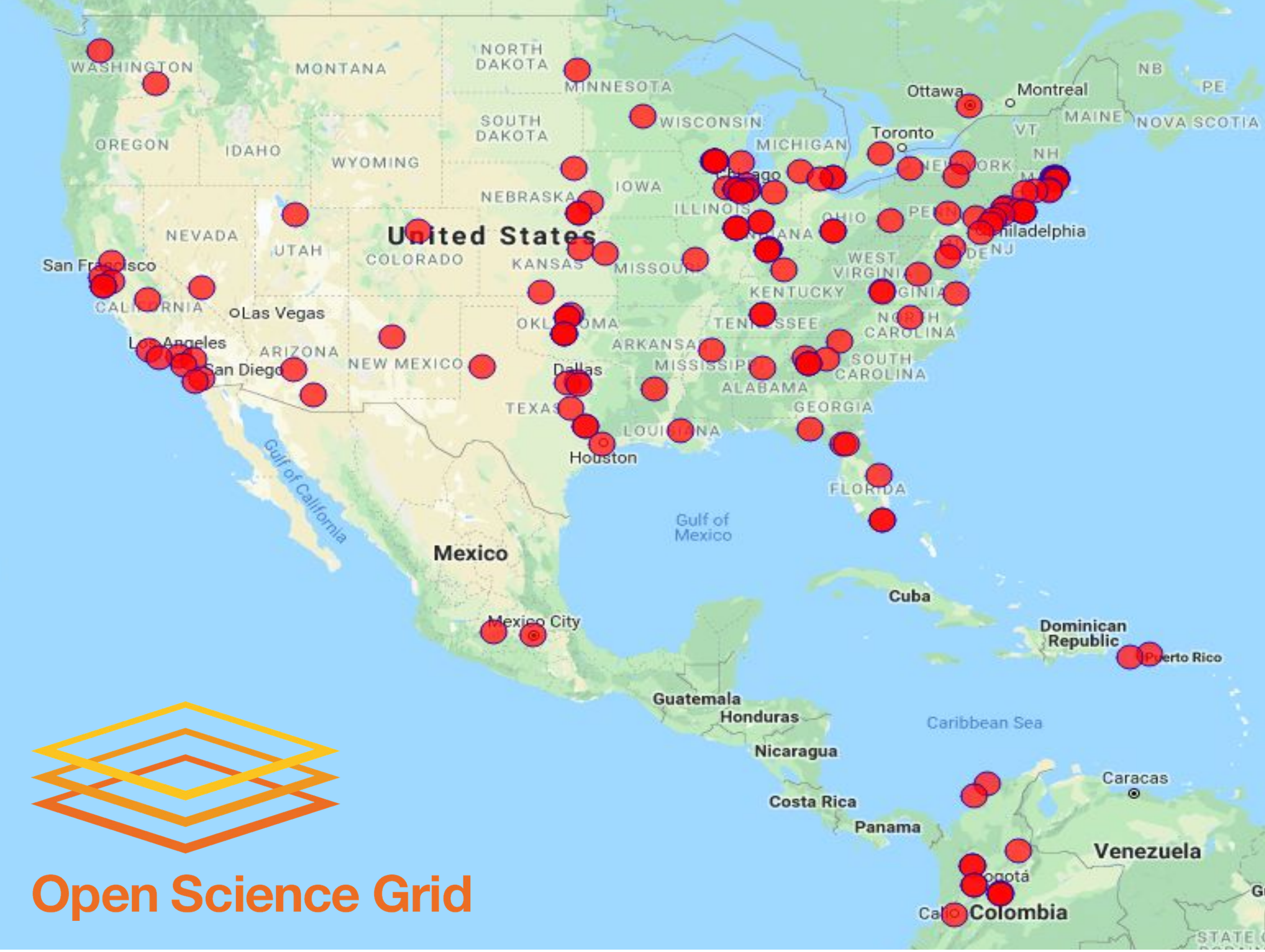
Introduction to OSG and Pegasus WMS

Mats Rynge, USC Information Sciences Institute

OSG Facilitation

What is the **Open Science Grid?**

In the last 24 Hours	
212,000	Jobs
4,800,000	CPU Hours
8,883,000	Transfers
1,172	TB Transfers
In the last 30 Days	
8,633,000	Jobs
138,645,000	CPU Hours
185,074,000	Transfers
25,452	TB Transfers
In the last 12 Months	
102,084,000	Jobs
1,634,482,000	CPU Hours
1,786,554,000	Transfers
277,000	TB Transfers



A consortium of researchers and institutions who <u>*share*</u> compute and data resources for ***distributed* high-throughput computing (<u>d</u>HTC)**

Open Science Grid

OSG serves 4 distinct groups

- The individual researchers and small groups on OSG-Connect
- The campus research support organizations
  - Teach IT organizations & support services so they can integrate with OSG
  - Train the Trainers (to support their researchers)
- Multi-institutional science teams
  - XENON, GlueX, SPT, Simons, and many many more
  - Collaborations between multiple campuses
- The 4 “big science” projects:
  - US-ATLAS, US-CMS, LIGO, IceCube

What is HTC?: An Analogy

Is it OSG-able?

Open Science Grid

Per-Job Resources	**Ideal Jobs!** (up to 10,000 cores, per user!)	**Still Very Advantageous!**	**Probably not…**
cores (GPUs)	**1** (1; non-specific)	**<8** (1; specific GPU type)	**>8 (or MPI)** (multiple)
Walltime *(per job)*	**<10 hrs*** *or checkpointable	**<20 hrs*** *or checkpointable	**>20 hrs**
RAM *(per job)*	**<few GB**	**<10 GB**	**>10 GB**
Input *(per job)*	**<1 GB**	**<10 GB**	**>10 GB**
Output *(per job)*	**<1 GB**	**<10 GB**	**>10 GB**
Software	*'portable' (pre-compiled binaries, transferable, containerizable, etc.)*	*most other than* □□□	*licensed software; non-Linux*

Open Science Grid

Federation = distributed control

OSG works on three simple principles:

1. Resource Owners determine policy of use
   - This means that all policy of use is set locally by the clusters that join the federation.
2. Resource Consumers specify the types of resources to use
   - How much RAM? How many cores per node? …
3. OSG submits its *own* batch system as ‘jobs’ into local batch systems.
   - User jobs are submitted locally, queued centrally, and execute anywhere that matches requirements after resource becomes available.

OSG operates overlay systems as services for all of science

What's Different about OSG?

1. **HTC is frequently new to users**
   - 'splitting up' work, optimizing throughput, etc.
   - many have HTC-able work and don't know

2. **OSG job logistics are different than using local resources**
   - file transfer vs. shared filesystems
   - software portability vs. system-wide installation
   - inherent interruption/retry
   - testing and troubleshooting on non-local resources

Open Science Grid

Open Science Grid

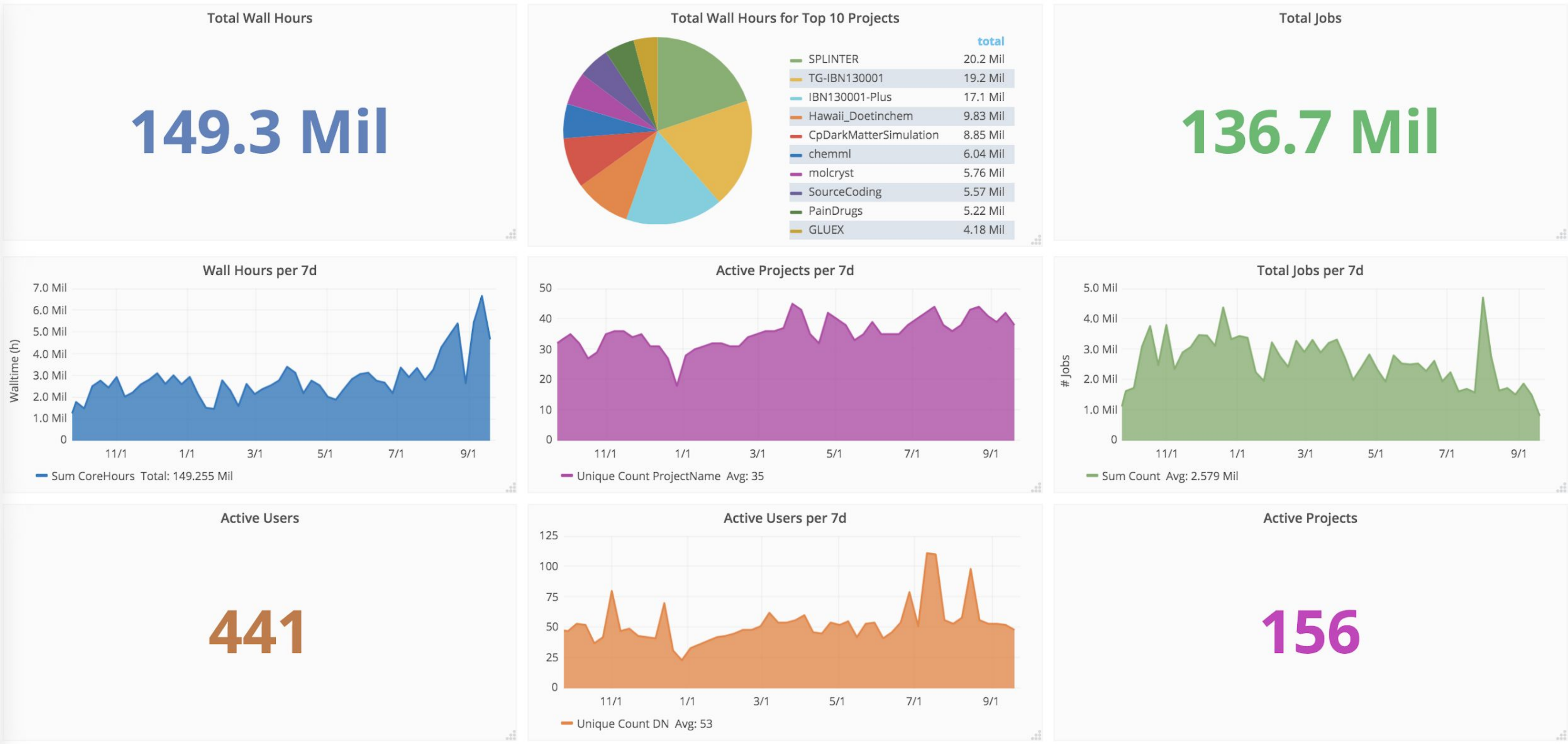
For individual researchers: **OSG Connect**

Access to and support for using OSG's open submission point

- **osgconnect.net > "Sign Up"**
- *available to researchers at any U.S. academic, government, or non-profit organization*
- includes:
  - initial consultation with an OSG Research Computing Facilitator
  - online documentation and examples
  - access to OSG's central software modules
  - (roughly) unlimited scratch; space for staging large input (Stash); built-in data caching

Open Science Grid

OSG Connect
(in the last year)



When to pursue an institutional/project submit node?

*… like having a cluster where you don't administer the 'worker' nodes, but **still provide all of the user support**.*

The institution is (at least) responsible for:

- user *facilitation*
  - incl. software portability (jobs *may* use OSG modules), troubleshooting
- administering user authorization
- (some) HTCondor administration on the submit node
- administering/integrating any institutional data storage

Open Science Grid

Pegasus Workflow Management System

USC Viterbi
School of Engineering
Information Sciences Institute

https://pegasus.isi.edu

Why Pegasus?

Automates complex, multi-stage processing pipelines

Enables parallel, **distributed computations**

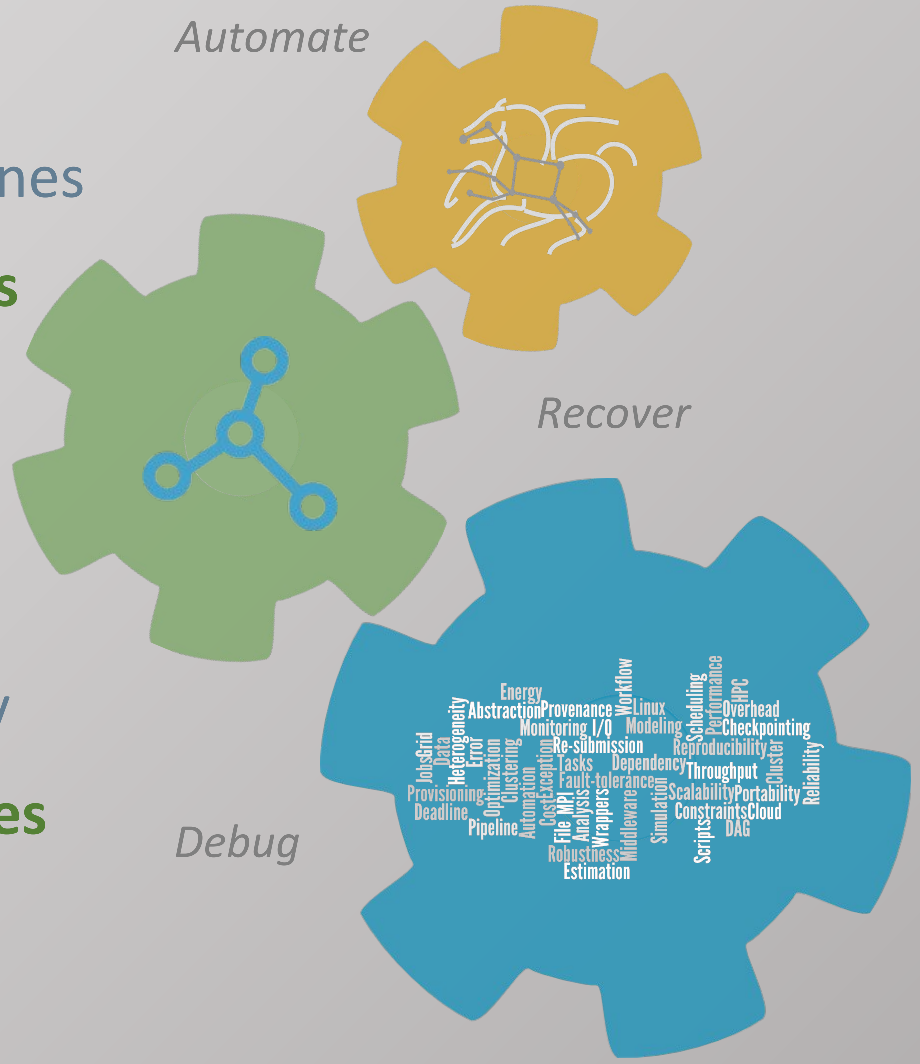
Automatically executes data transfers

Reusable, aids **reproducibility**

Records how data was produced (**provenance**)

Handles **failures** with to provide reliability

Keeps track of data and **files**



HTCondor
High Throughput Computing

NSF funded project since 2001, with close collaboration with HTCondor team

Advanced LIGO PyCBC Workflow

One of the main pipelines to measure the statistical significance of data needed for discovery

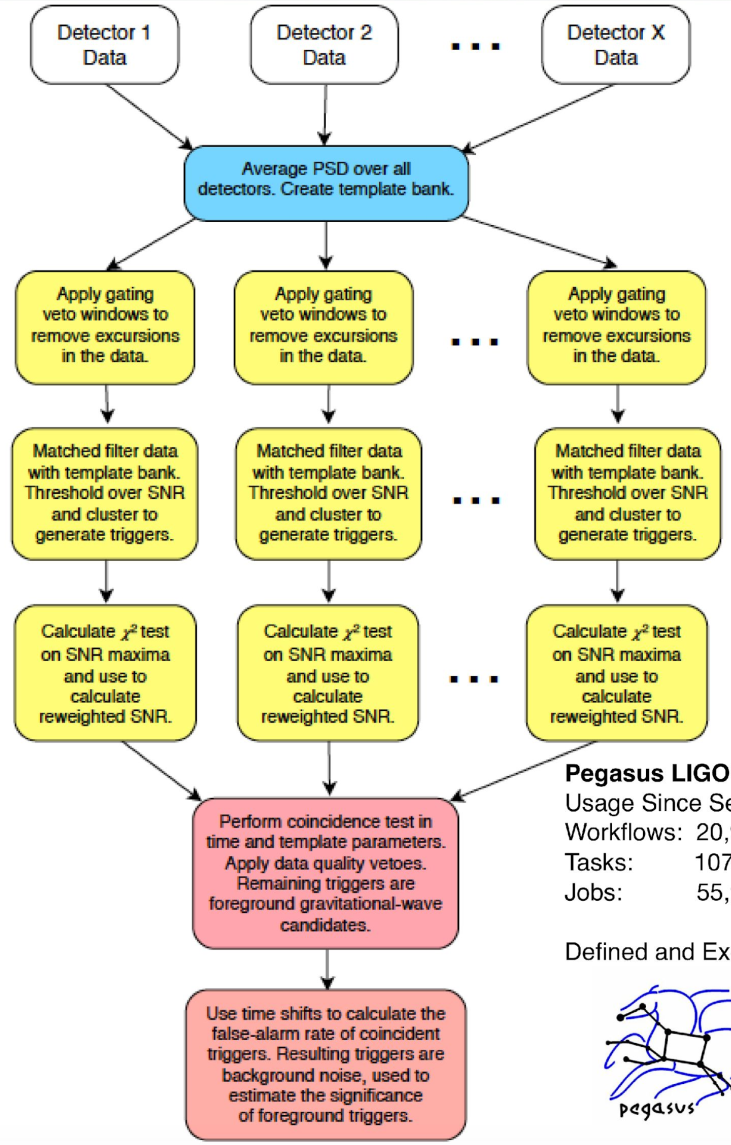
Contains **100,000s of jobs** and accesses on order of **terabytes of data**

Uses data from multiple detectors

A single run of the binary black hole + binary neutron star search through the O1 data (about 3 calendar months of data with 50% duty cycle) requires a **workflow** with **194,364 jobs**

Generating the final O1 results with all the review required for the first discovery took about **20 million core hours**

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



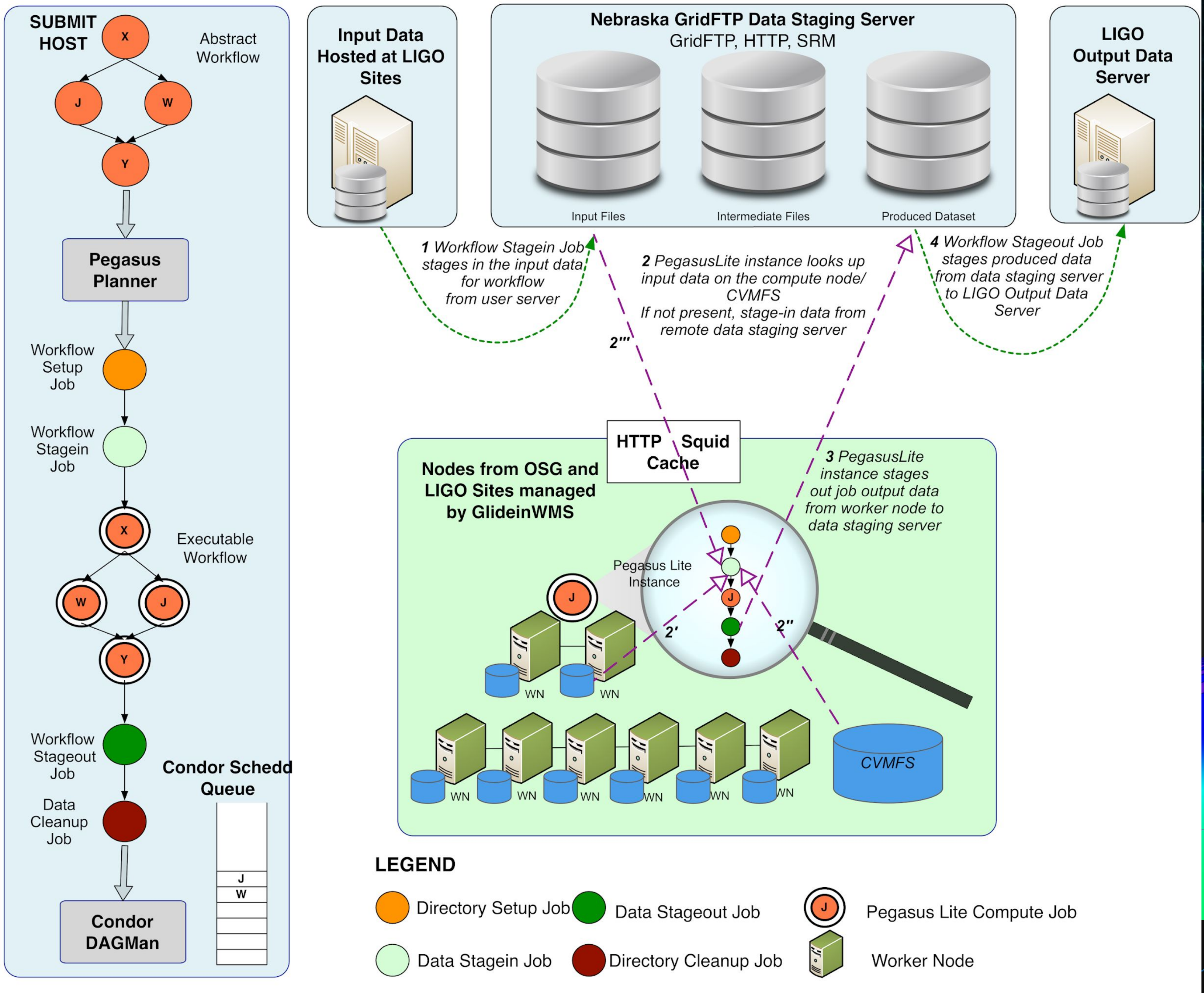
Pegasus LIGO PyCBC Workflow
Usage Since Sept 2015
Workflows: 20,942
Tasks: 107,576,294
Jobs: 55,915,928

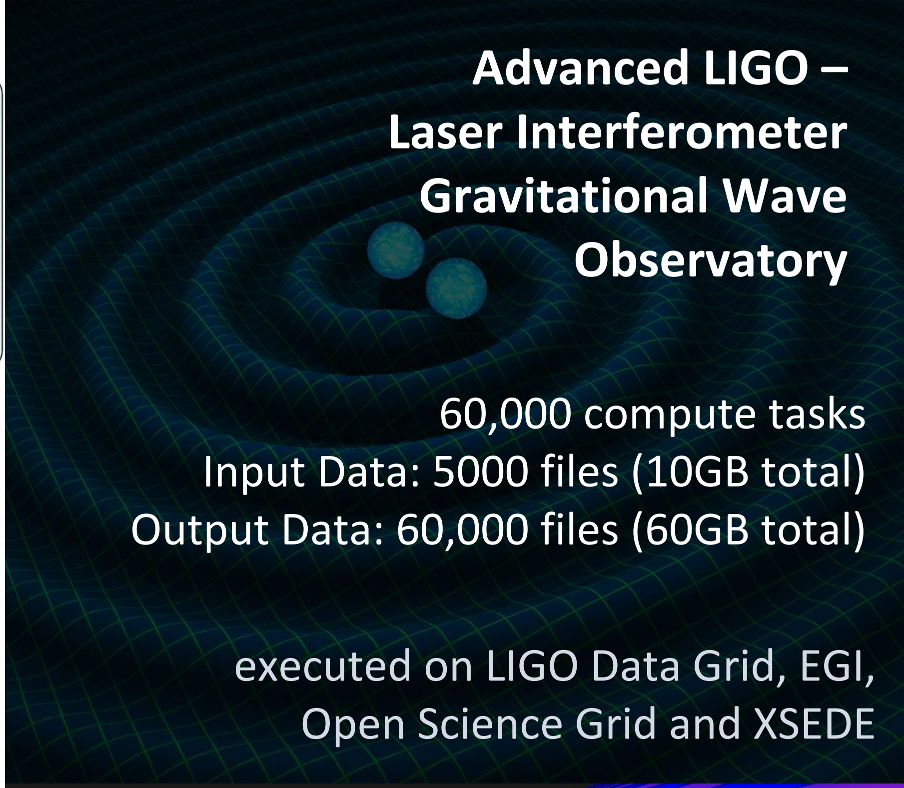
Defined and Executed by Pegasus

Pegasus

PyCBC Papers: An improved pipeline to search for gravitational waves from compact binary coalescence. *Samantha Usman, Duncan Brown et al.*
The PyCBC search for gravitational waves from compact binary coalescence, *Samantha Usman et al* (https://arxiv.org/abs/1508.02357)
PyCBC Detection GW150914: First results from the search for binary black hole coalescence with Advanced LIGO. *B. P. Abbott et al.*

Data Flow for LIGO Pegasus Workflows in OSG

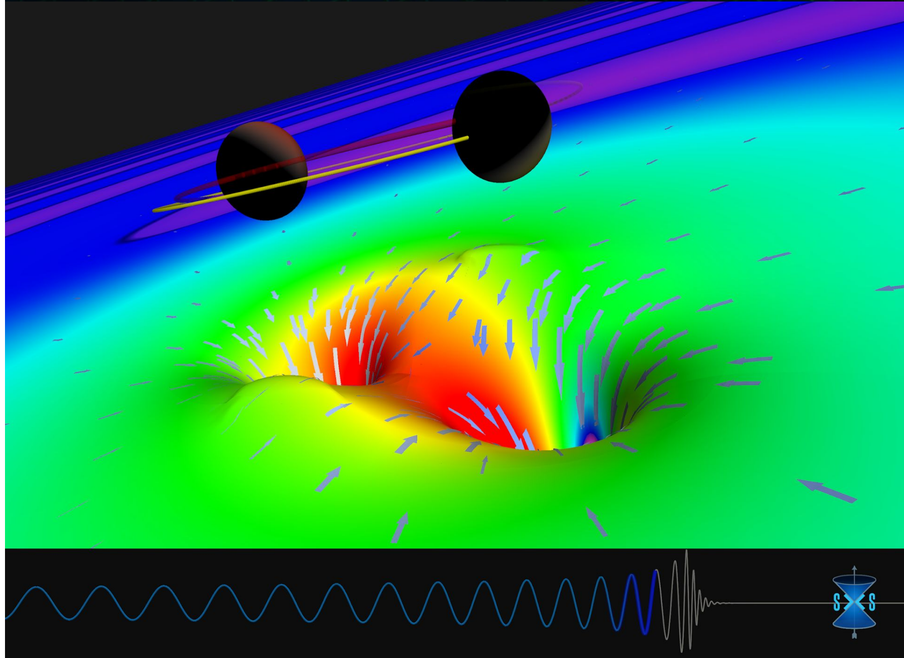




Advanced LIGO – Laser Interferometer Gravitational Wave Observatory

60,000 compute tasks
Input Data: 5000 files (10GB total)
Output Data: 60,000 files (60GB total)

executed on LIGO Data Grid, EGI, Open Science Grid and XSEDE



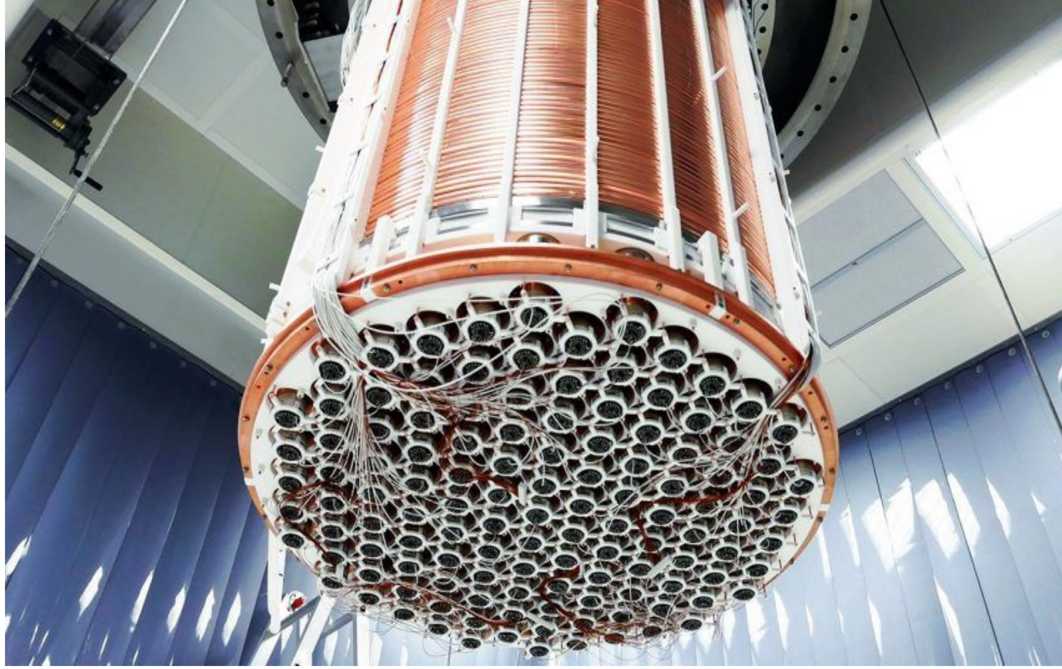
XENONnT - Dark Matter Search

Two workflows: Monte Carlo simulations, and the main processing pipeline.

Workflows execute across Open Science Grid (OSG) and European Grid Infrastructure (EGI)

Rucio for data management

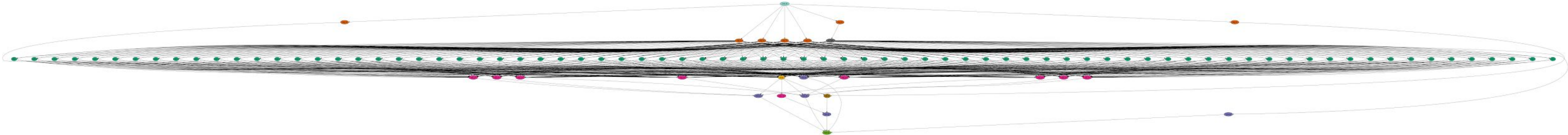
MongoDB instance to track science runs and data products.



```
------------------------------------------------------------------------------
Type           Succeeded Failed  Incomplete  Total     Retries   Total+Retries
Tasks          4000      0       0           4000      267       4267
Jobs           4484      0       0           4484      267       4751
Sub-Workflows  0         0       0           0         0         0
------------------------------------------------------------------------------

Workflow wall time                                       : 5 hrs, 2 mins
Cumulative job wall time                                 : 136 days, 9 hrs
Cumulative job wall time as seen from submit side        : 141 days, 16 hrs
Cumulative job badput wall time                          : 1 day, 2 hrs
Cumulative job badput wall time as seen from submit side : 4 days, 20 hrs
```

Main processing pipeline is being developed for XENONnT - data taking will start at the end of 2019. Workflow in development:

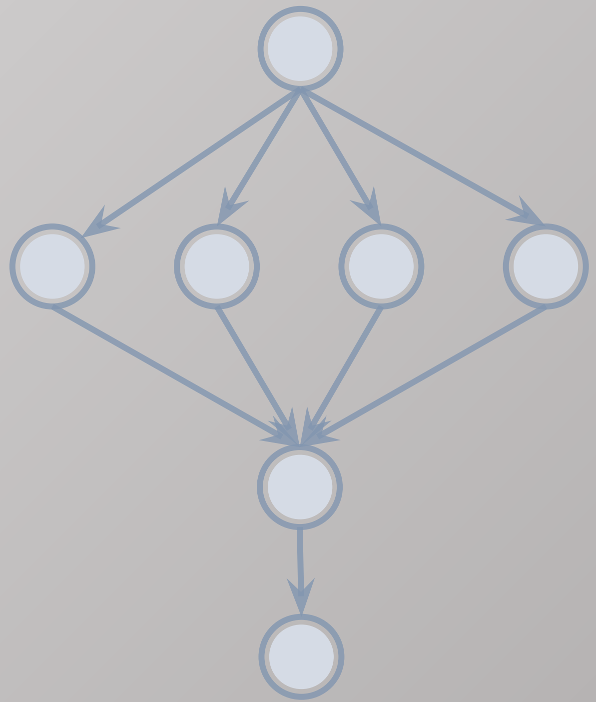


Key Pegasus Concepts

HTCondor High Throughput Computing

Pegasus WMS == Pegasus planner (mapper) + DAGMan workflow engine + HTCondor scheduler/broker

- Pegasus maps workflows to infrastructure
- DAGMan manages dependencies and reliability
- HTCondor is used as a broker to interface with different schedulers



Workflows are DAGs

- Nodes: jobs, edges: dependencies
- No while loops, no conditional branches
- Jobs are standalone executables

Planning occurs ahead of execution

Planning converts an abstract workflow into a concrete, executable workflow

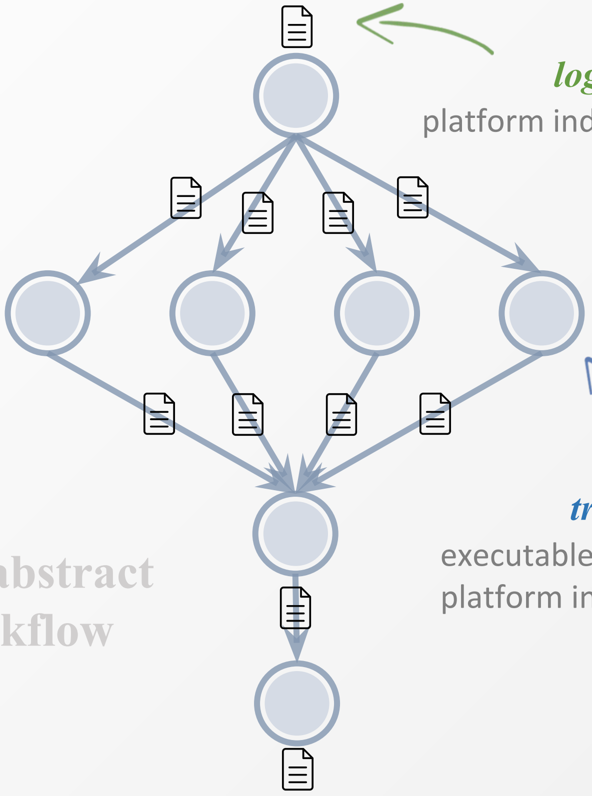
- Planner is like a compiler

DAX
DAG in XML

DAG
directed-acyclic graphs

Portable Description

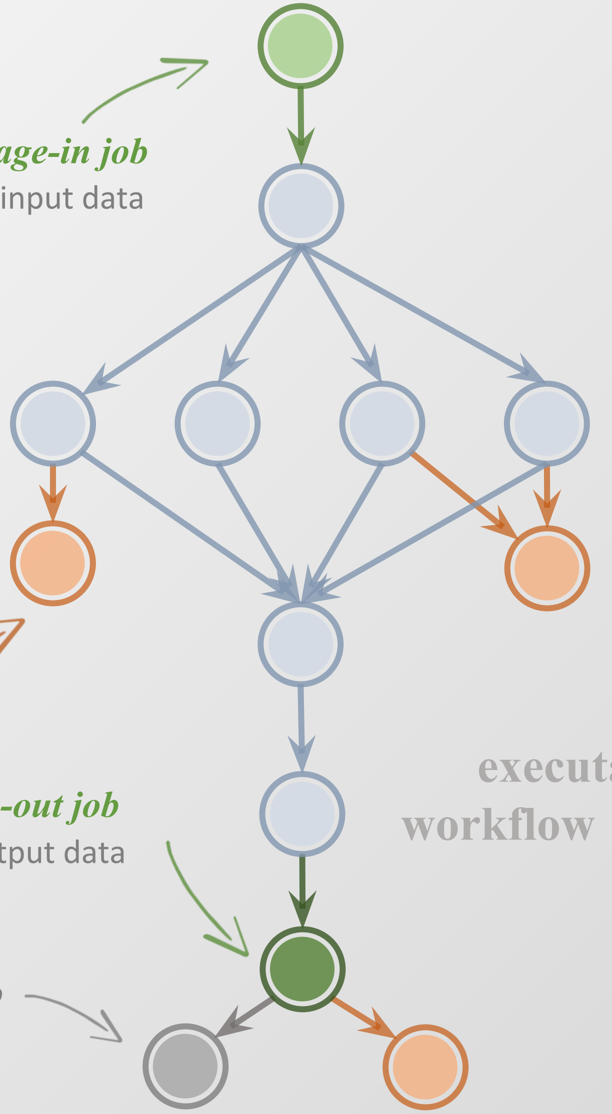
Users do not worry about
low level execution details



logical filename (LFN)
platform independent (abstraction)

transformation
executables (or programs)
platform independent

abstract workflow



stage-in job
Transfers the workflow input data

cleanup job
Removes unused data

stage-out job
Transfers the workflow output data

registration job
Registers the workflow output data

executable workflow

Challenges from the pov of a infrastructure/software providers

Access to cycles / running jobs is now the easy part! Challenges are elsewhere:

Software Environments

- Many solutions are available, but few projects are “formal” about their environments
- Reproducibility
- Underestimation of effort

Data Management

- Tracking, Life cycle, Disaster recovery, Versioning

Workflows

- Data access/movement
- Data integrity
- Provenance
- Reproducibility

Open Science Grid

Submit locally, run globally.

Questions?

help@opensciencegrid.org