Scientific Workflows with Pegasus

Karan Vahi

Information Sciences Institute,
University of Southern California
vahi@isi.edu
Scientific Workflows

• An abstraction to express ensemble of complex computational operations
  - *Eg: retrieving data from remote storage services, executing applications, and transferring data products to designated storage sites*

• A workflow is represented as a directed acyclic graph (DAG)
  - *Nodes: tasks or jobs to be executed*
  - *Edges: depend between the tasks*

• Have a monolithic application/experiment?
  - *Find the inherent DAG structure in your application to convert into a workflow*
Workflow Challenges Across Domains

- Describe complex workflows in a simple way
- Access distributed, heterogeneous data and resources (heterogeneous interfaces)
- Deal with resources/software that change over time
- Ease of use. Ability to debug and monitor large workflows

Our Focus

- Separation between workflow description and workflow execution
- Workflow planning and scheduling (scalability, performance)
- Task execution (monitoring, fault tolerance, debugging, web dashboard)
- Provide additional assurances that a scientific workflow is not accidentally or maliciously tampered with during its execution.

Pegasus is a NSF funded project since 2001, with close collaboration with HTCondor team
Key Pegasus Concepts

▲ **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
Portable Description
Users do not worry about low level execution details

**Input Workflow Specification** YAML formatted

**Output Workflow**
directed-acyclic graphs

**Logical Filename (LFN)**
platform independent (abstraction)

**Transformation**
Executables (or programs)
platform independent

**Stage-in Job**
Transfers the workflow input data

**Cleanup Job**
Removes unused data

**Stage-out Job**
Stage-out generated output data

**Registration Job**
Registers the workflow output data

https://pegasus.isi.edu
Performance.

Why not improve it?

Clustered Job
Groups small jobs together to improve performance

Task
Small granularity
Pegasus provides tools to generate the Abstract Workflow

```python
#!/usr/bin/env python3

import os
import logging
from pegasus.api import *

logging.basicConfig(level=logging.DEBUG)

# -- Import Pegasus API -------------------
from pegasus.api import *

# -- Create Abstract workflow -----------
wf = Workflow('pipeline')

# -- Create Abstract workflow -----------
workflow = File('pegasus.html')

# -- Create Abstract workflow -----------
curl_job = {
    .add_args('-o', workflow, 'https://pegasus.isi.edu/')
    .add_output(workflow, stage_out=True, register_replica=True)
}

# -- Create Abstract workflow -----------
mi_job = {
    .add_args('-i', workflow)
    .add_output(workflow, stage_out=True, register_replica=True)
}

# -- Add jobs to the Abstract Workflow ---
wf.add_jobs(curl_job, mi_job)

# -- Add control flow dependency ------
wf.add_dependency(mi_job, parents=[curl_job])

# -- Write out the Abstract Workflow ----
wf.writedir()
```

**YAML Formatted**

```
x-pegasus:
  app_lang: python
  created_by: vohei
  created_on: 11-19-2021 14:57:58Z
  pegasus: "5.0"
  name: pipeline
  jobs:
  - type: job
    name: curl
    arguments:
      - -o
      - pegasus.html
      - http://pegasus.isi.edu
      - pegasus.html
      - stdin: count.txt
      - arguments:
        - -i
        - pegasus.html
        type: output
        stageOut: false
        registerReplica: false
      - type: job
        name: wc
        stdin: count.txt
        arguments:
        - -i
        - pegasus.html
        type: output
        stageOut: true
        registerReplica: true
        - type: job
          stdin: pegasus.html
          type: input
          stageOut: true
          registerReplica: true
  jobDependencies:
  - id: 10000000
    children:
    - 10000001
```

Abstract Workflow
Pegasus Deployment

- **Workflow Submit Node**
  - Pegasus WMS
  - HTCondor

- **One or more Compute Sites**
  - Compute Clusters
  - Cloud
  - OSG

- **Input Sites**
  - Host Input Data

- **Data Staging Site**
  - Coordinate data movement for workflow

- **Output Site**
  - Where output data is placed
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.
$ pegasus-status pegasus/examples/split/run0001
STAT IN_STATE JOB
Run 00:39 split-0 (/home/pegasus/examples/split/run0001)
Idle 00:03 split_ID0000001
Summary: 2 Condor jobs total (I:1 R:1)
UNRDY READY PRE IN_Q POST DONE FAIL %DONE STATE   DAGNAME
14 0 0 I 0 2 0 11.8 Running *split-0.dag

$ pegasus-analyzer pegasus/examples/split/run0001
pegasus-analyzer: initializing...

************************Summary***************************
Total jobs : 7 (100.00%)
# jobs succeeded : 7 (100.00%)
# jobs failed : 0 (0.00%)
# jobs unsubmitted : 0 (0.00%)

$ pegasus-statistics -s all pegasus/examples/split/run0001
------------------------------------------------------------------------------
<table>
<thead>
<tr>
<th>Type</th>
<th>Succeeded</th>
<th>Failed</th>
<th>Incomplete</th>
<th>Total</th>
<th>Retries</th>
<th>Total+Retries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasks</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Jobs</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>Sub-Workflows</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
------------------------------------------------------------------------------
Workflow wall time : 2 mins, 6 secs
Workflow cumulative job wall time : 38 secs
Cumulative job wall time as seen from submit side : 42 secs
Workflow cumulative job badput wall time :
Cumulative job badput wall time as seen from submit side :

Provenance Data can be Summarized

Or

Used for Debugging

https://pegasus.isi.edu
Pegasus also handles large-scale workflows

Sub-Workflow

Recursion ends When abstract workflow with only compute jobs is encountered

https://pegasus.isi.edu
Automatic Integrity Checking in Pegasus

Pegasus performs integrity checksums on input files right before a job starts on the remote node.

- For raw inputs, checksums specified in the input replica catalog along with file locations
- All intermediate and output files checksums are generated and tracked within the system.
- Support for sha256 checksums

Job failure is triggered if checksums fail
Pegasus is part of the ACCESS support strategy

Pegasus is be used as a tier 1 tool

Central Open OnDemand instance with Pegasus, HTCondor and Jupyter

It is be easy to run HTC workflows across ACCESS sites
ACCESS Pegasus

Bring your workflows to ACCESS!

- Execute scientific workflows across ACCESS resources

- OpenOnDemand Portal: **has all you need**: Jupyter Notebooks, ACCESS authentication, Pegasus workflow management, and HTCondor job management

- **Bring your own ACCESS capacity**: HTCondor Annex - pilot jobs automatically create a virtual HTCondor pool

https://access.pegasus.isi.edu

More at: support.access-ci.org/pegasus
Southern California Earthquake Center’s CyberShake

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

First Physics-Based “Shake map” of Southern California

Laser Interferometer Gravitational-Wave Observatory (LIGO)

High-throughput computing workload, access to HPC resources, ~ 21k Pegasus workflows, ~ 107M tasks
Supported since 2001, distributed data, opportunistic computing resources

First direct detection of a gravitational wave (colliding black holes)

XENONnT - Dark Matter Search

Custom data management
Rucio for data management
MongoDB instance to track science runs and data products.

Monte Carlo simulations and the main processing pipeline.
Event Horizon Telescope
Bringing Black Holes into Focus

8 telescopes: 5 PB of data
60 simulations: 35 TB data

First images of black hole at the center of the M87 galaxy

Improve constraints on Einstein’s theory of general relativity by 500x

480,000 jobs - 2,600,000 core hours
#15 in all OSG projects in last 6 months
#2 in all OSG astronomy projects in the last 6 months

Pegasus-SYMBa Pipeline

Physically accurate synthetic observation data from simulations are keys to develop calibration and imaging algorithms, as well as comparing the observation with theory and interpreting the results.

1. Pull simulated image from CyVerse Data Store
2. Model interstellar scattering
3. Synthetic interferometry data
4. Model observation systematics
5. Calibrate synthetic observation
6. Push synthetic observation data back to CyVerse

https://pegasus.isi.edu
Processing instrument data in real time
Automated Quality Control of Phenotypic Datasets

The NIMH Center for Collaborative Genomic Studies on Mental Disorders, now known as the NIMH Repository and Genomics Resource (NRGR), maintains biomaterials, demographic, and phenotypic data from over 200,000 well-characterized individuals with a range of psychiatric illnesses, their family members, and unaffected controls.

- Easy to Use Web-Based Interface
  - Simple Submission
  - Real-time Monitoring and Error Reports
  - After automated QC, submit corrected files for expert curation

- Scalable
  - Workflow based architecture using Pegasus WMS

- Extensible Design
  - Easily add new QC steps, and checks

- Enables Complex checks
  - Pedigree Checks
  - QC Checks validating data with external sources
  - QC Checks can correlate data across multiple files and across multiple fields within files

- Ensures high-quality uniform data deposited at NRGR
- Better resource utilization: solve most QC problems automatically, use expert curation for hard cases

https://pegasus.isi.edu
Get Started

- **Pegasus Website**
  https://pegasus.isi.edu

- **Users Mailing List**
  pegasus-users@isi.edu

- **Support**
  pegasus-support@isi.edu

- **Slack**
  Ask for an invite by trying to join pegasus-users.slack.com in the Slack app

- **Pegasus Online Office Hours**
  https://pegasus.isi.edu/blog/online-pegasus-office-hours/

**YouTube Channel**

https://www.youtube.com/channel/UCwJQln1CqBvTJqiNr9X9F1Q/featured

**Bi-monthly basis on second Friday of the month, where we address user questions and also apprise the community of new developments**

https://pegasus.isi.edu