ACCESS Pegasus

A hosted scientific workflow system, part of the ACCESS Support approach

Mats Rynge, USC/ISI
About Me

- Computer scientist at USC Information Sciences Institute
- Workflow developer
- OSG Facilitator
- Formerly ECSS
- ... and now ACCESS Support
Overview

Scientific Workflows
ACCESS Support
Logging In
Setting Up Resources
Designing Workflow
Provisioning Resources
Monitoring Workflow and Resources
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*
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
**Advanced LIGO** in OSG

- **Data Flow for LIGO Workflows**

- **60,000 Compute Tasks**
- **Output Data: 60,000 files (60GB total)**
- **Processed Data: 7.25GB**

**LEGEND**
- Directory Setup Job
- Data Stagein Job
- Data Cleanup Job
- Worker Node
- Pegasus Lite Compute Job
- Pegasus Lite Stagein Job
- Pegasus Lite Stageout Job
- Data Cleanup Job
- Data Stageout Job
- Directory Setup Job

**Processes:**
1. Workflow Stagein Job stages in the input data for execution on the worker node.
2. Workflow Stageout Job stages out the produced data from the worker node to the data staging server.
3. Pegasus Lite instance stages in the input data to the worker node.
4. Pegasus Lite instance stages out the produced data to the LIGO output data server.

**Sites:**
- LIGO Sites
- Nebraska GridFTP Data staging server
- LIGO output data server

**Networks:**
- GridFTP, HTTP, SRM

**Data Staging:**
- Input data hosted at LIGO sites

**Data Workflow:**
- Abstract Workflows
- Pegasus Planner
- Condor DAGMan
- Workflow Setup
- Workflow Stagein
- Workflow Stageout
- Executable
- Data Cleanup

**Summary:**
- Input Data: 5000 files (10GB total)
- Output Data: 60,000 files (60GB total)
NSF ACCESS Program Structure

ACCESS Services

Allocations
- Allocation Services
- Innovative Pilots
- Service Model

Support
- General User Assistance
- Allocation & Utilization Assistance
- End User Training
- Computational Science Support Network

Operations
- Operational Support
- Data & Networking Support
- Cybersecurity Support

Metrics
- M&O Operations
- Service Model
- Data Analytics Framework

ACCESS Coordination Office Services

Executive Council

Communications & Outreach

Advisory Board
Enable innovative research through equitable and scalable support

Four tiers of support

Tools, growing knowledge base

Match-making with experts

Student engagement

Engagement from community

CSSN incentives

[Diagram showing four tiers of support:

- TIER 1: Easy to Use Tools - ACCESS ONDEMAND & PEGASUS
- TIER 2: Curated Knowledge Base
- TIER 3: MATCH-Plus - SHORT-TERM SUPPORT PARTNERSHIPS
- TIER 4: MATCH-Premier - LONG-TERM EMBEDDED SPECIALISTS]

https://support.access-ci.org/
ACCESS OnDemand Pilot

ACCESS OnDemand (AOD) Goals:
1. Improve productivity for researchers by integrating Open OnDemand with ACCESS services (e.g., authentication, support portal, and metrics)
2. Decrease effort for resource providers to install, configure, and update OnDemand through managed releases with accompanying integration roadmaps

Pilot Project
1. Initial AOD pilot targeting current OnDemand deployments on ACCESS resources (e.g., Anvil, Bridges 2, Delta and Expanse)
2. Working group is forming! Please contact Dave Hudak (dhudak@osc.edu) for details.

https://support.access-ci.org/ondemand
Opportunities for Engagement

• ACCESS CSSN Community Participation Program (CCEP)
  • Earn travel support rewards by contributing to the ACCESS Knowledge Base, joining committees, becoming a mentor
• Advisory committees – forming now!
  • CCEP Committees
  • MATCH-Plus/MATCH-Premier Steering Committees
• Join our mailing list and apply for the CCEP!
  https://support.access-ci.org/cssn
• Pegasus, OnDemand, MATCH-Plus and MATCH-Premier pilots – opportunities for researchers, resource providers, consultants, mentors and students!
• Listening tours - we need your feedback!
https://support.access-ci.org/pegasus
Prepare

Logging In

CILogin with your ACCESS ID and institutional login

- https://access.pegasus.isi.edu

All registered ACCESS users with an active allocation automatically have access.
Prepare

Setting Up Resources

One time setup

Use Open OnDemand instances at resource providers to install ssh keys, and determine local allocation id
Step 1

Designing Workflow

Pegasus API in Jupyter Notebook

Fully hosted environment, based on Open Ondemand
Step 2

Provision Resources

Use the HTCondor Annex tool to dynamically bring in compute nodes from one or more resource providers.
HTCondor Annex / Pilot Jobs

- A pilot can run multiple user jobs - it stays active until no more user jobs are available or until end of life has been reached, whichever comes first.

- A pilot is partitionable - job slots will dynamically be created based on the resource requirements in the user jobs. This means you can fit multiple user jobs on a compute node at the same time.

- A pilot will only run jobs for the user who started it.
Step 3
Monitors Workflow and Resources

Workflows can be monitored from within the Jupyter notebook, or via command line.

HTCondor Annex can be monitored on the command line:

```
[rynge@access ~]$ htcondor annex
status $USER
```

Note: the line in the output that starts with pegasus-status contains the relevant information.
Try it out!

Documentation:

https://support.access-ci.org/pegasus

Open a ticket:

https://support.access-ci.org/open-a-ticket

Questions?