Leveraging Pegasus 4.0 and GlideinWMS for Executing Data Intensive Workflows on OSG

Karan Vahi, Mats Rynge, Gaurang Mehta, Rajiv Mayani, Jens Vöckler, , Ewa Deelman
Information Sciences Institute, University of Southern California

Overview

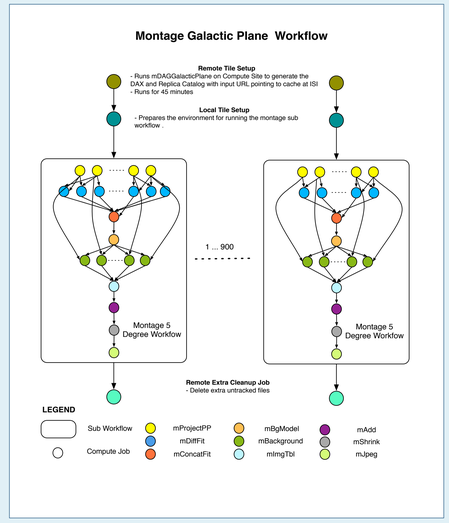
- Pegasus is a system for mapping and executing abstract application workflows over a range of execution environments.
- The output is an executable workflow that can be executed over a variety of resources (Clouds, XSEDE, OSG, Campus Grids, Clusters, Workstation)
- Pegasus can run workflows comprising of millions of tasks.
- Pegasus Workflow Management System (WMS) consists of three main components: the Pegasus mapper, Condor DAGMan, and the Condor schedd.
- The mapping of tasks to the execution resources is done by the mapper based on information derived from static and/or dynamic sources. Pegasus adds and manages data transfer between the tasks as required.
- DAGMan takes this executable workflow and manages the dependencies between the tasks and releases them to the Condor schedd for execution.

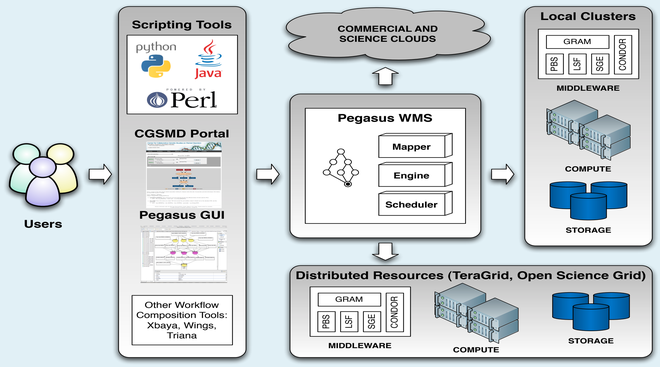
Pegasus 4.0 improvements for running on GlideinWMS

- GlideinWMS provides an excellent dynamic execution environment for Pegasus workflows
- Pegasus 4.0 introduces new advanced data handling capabilities
- Contains improved support for running workflows in non-shared filesystem scenarios such as on top of GlideinWMS.
- Pegasus now optionally separates the data staging site from the workflow execution site for more flexible data management.
- A new feature is PegasusLite - an autonomous lightweight execution environment to manage jobs on the compute nodes and handles data movement to/from such jobs against the workflow staging site
- Pegasus 4.0 is currently available on the OSG XSEDE and OSG Engagement glideinWMS submit nodes

Large Scale Hierarchal Workflows

- Nodes in a workflow can be tasks or another workflow (DAX).
  - Scales up-to order of millions of tasks
- Each sub workflow is mapped when it is ready for execution.



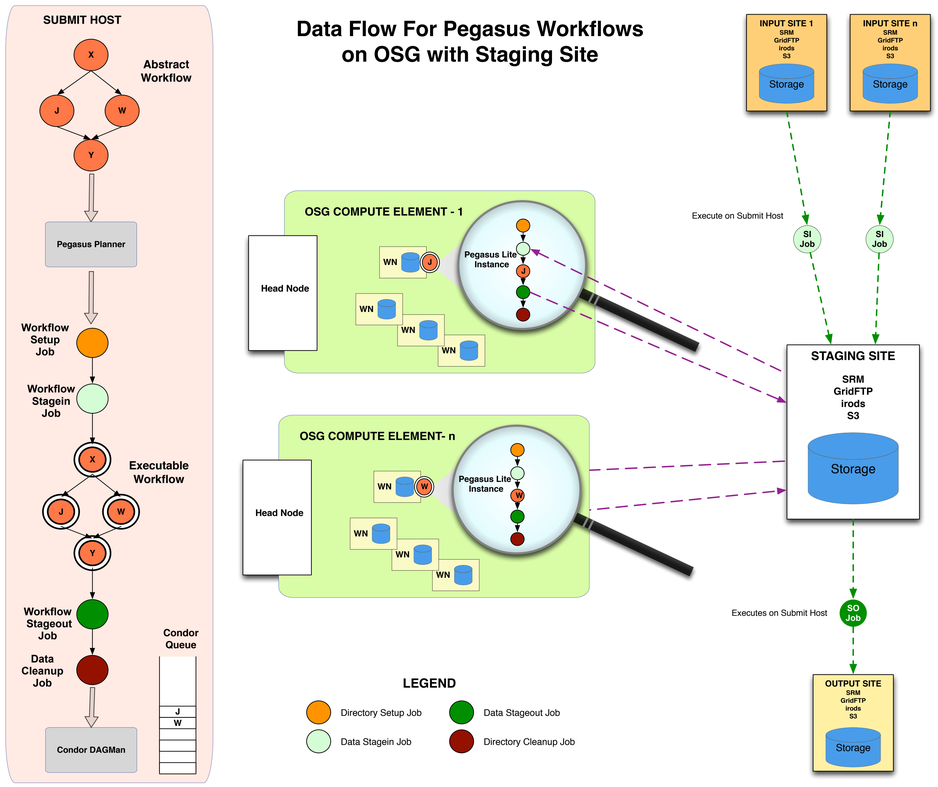


Pegasus Features

- Clustering of small tasks into large clusters for performance.
- Optimized data transfers and ability to use different protocols.
- Data reuse in case intermediate data products are available
  - workflow-level checkpointing
- Automatic data cleanup
  - reduces data footprint
- Support for Workflow and Task level notifications
- Integrates with Resource Provisioners like GlideinWMS.
- Support for Shell Code Generator

Data Staging Configurations Supported by Pegasus

- **Shared Filesystem** (Head Node and the worker nodes of execution sites share a filesystem)
- **Non Shared Filesystem with Staging Site** (Head Node and Worker Nodes don't share a filesystem). Data is staged from an external staging site
- **CondorIO** (Head Node and Worker Nodes don't share a filesystem).Data is staged from the submit host using Condor File Transfers



Data Flow For a Workflow with Pegasus on OSG with Staging Site

1. Workflow Stagein Jobs transfer input data for the workflow to the staging site
2. Pegasus Lite wrapped jobs , when they start on OSG worker nodes, pull in the input data from staging site
3. The compute job executes on a local directory on the worker node.
4. The PegasusLite wrapper pushes the output data from the worker node back to the staging site
5. The Workflow Stageout Jobs transfer the relevant output data out to the output site from staging site

Galactic Plane Workflow

Astronomy and Physics

- Galactic Plane for generating mosiacs from the Spitzer Telescope
- Used to generate tiles 360 x 40 around the galactic equator
- A tile 5 x 5 with 1 overlap with neighbors
- Output datasets to be used in NASA Sky and Google Sky
- One workflow run for each of 17 bands (wavelengths)
- Each sub workflow uses 3.5TB of input imagery (1.6 million files)
- Each workflow consumes 30K CPU hours and produces 900 tiles in FITS format

Proposed Runs on Xsede and OSG

- Run workflows corresponding to each of the 17 bands
- Total Number of Data Files – 18 million
- Potential Size of Data Output – 86 TB

Monitoring and Debugging Capabilities

- Workflow Progress can be tracked through a database.
- Stores provenance of data used, produced and which software was used with what parameters
- Retries computations in case of failures.
- Monitoring and Debugging tools to debug large scale workflows.

Acknowledgments:

- Pegasus WMS is funded by the National Science Foundation OCI SDCI program grant #0722019.
- Condor : Miron Livny, Kent Wenger, University of Wisconsin Madison

http://pegasus.isi.edu