Pegasus WMS: Enabling Large Scale Workflows on National Cyberinfrastructure

Karan Vahi, Ewa Deelman, Gideon Juve, Mats Rynge, Rajiv Mayani, Rafael Ferreira da Silva
University of Southern California / Information Sciences Institute

Overview

- 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 to 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.
- Pegasus Workflow Management System (WMS) consists of three main components: the Pegasus Mapper, HTCondor DAGMan, and the HTCondor Scheduler.
- XSEDE Tutorial: [https://sites.google.com/site/xsede/](https://sites.google.com/site/xsede/)

Workflow Design and Mapping

**DAX Generator API**

Easy to use APIs in Python, Java and Perl to generate an abstract workflow describing the users' computation.

Above is a simple two node hello world example.

**Abstract Workflow (DAX)**

The abstract workflow rendered as XML. It only captures the computations the user wants to do and is devoid of any physical paths. Input and output files are identified by logical identifiers. This representation is portable between different execution environments.

**Abstract to Executable Workflow (Condor DAG) Mapping**

The DAX is passed to the Pegasus Mapper and it generates an HTCondor DAGMan workflow that can be run on actual resource.

The above example highlights addition of data movement nodes to staging in the input data and stage out the output data; addition of data cleanup nodes to remove data that is no longer required; and registration nodes to catalog output data locations for future discovery.

Data Reuse Example

Additional Capabilities Highlighted

- **Data Reuse:** Jobs B and D are removed from the workflow as file f.d already exists. The f.d is staged in, instead of regenerating it by executing jobs B and D.
- **Job Clustering:** Jobs C and E are clustered together into a single clustered job.
- **Cross Site Run:** Single Workflow can be executed on multiple sites, with Pegasus taking care of the data movement between the sites.

Advanced LIGO pyCBC Workflows

- **Advanced LIGO pyCBC pipeline**
  - A single stage pipeline for analyzing data from various LIGO and VIRGO detectors.
  - Designed to search for gravitational waves from compact object binaries containing neutron stars and stellar-mass black holes that have been performed.
  - Actual runs on real data expected to start in September 2015.
  - Uses Pegasus WMS to run on XSEDE, LIGO Data Grid and OSG resources.

Test Runs on TACC Stampede with Pegasus:

- Uses Pegasus MPI Cluster to manage parallel FFT jobs into large clusters (using 256, 512 and 1024 cores) submitted to the SLURM batch queue via Globus GRAM.
- Task affinity is set in pegasus-mpi-cluster so that the threads for the FFTs for a single job stay pinned to a single processor to obtain optimal user of the CPU's L3 cache during execution of the FFT.
- Pegasus stages the outputs back to LIGO Data Grid for post processing.

Pegasus Workflows with PMC on XSEDE

- **XSEDE Challenges**
  - HPC resources with remote job submission via GRAM, with strict queue limits per user (~50)
  - Workflows can have a large number of tasks that cannot be all submitted through remote job submission interface

**Solution:**

- The workflow is partitioned into independent sub-graphs, which are submitted as self-contained Pegasus MPI Cluster (PMC) jobs to the remote sites.
- PMC relies on standard MPI constructs.

A PMC job is expressed as a DAG and PMC uses the MPI master-worker paradigm to farm out individual tasks to worker nodes. PMC acts a scheduler and considers core and memory requirements of the tasks when making scheduling decisions. PMC can be easier to setup than pilot jobs / HTCondor guidelines as no special networking is required.

Acknowledgments:

- Pegasus WMS is funded by the National Science Foundation OCI SDCI program grant #1148515.
- HTCondor: Miron Livny, Kent Wenger, University of Wisconsin Madison
- LIGO: Larne Pekowsky, Duncan Brown, Alexander H. Nitz and others from LIGO CBC group

http://pegasus.isi.edu