

Ghub Pegasus WMS Workflow with MATLAB Example

- Ghub scientific gateway is built on the HUBzero platform for scientific collaboration.
- Ghub provides you and other members of the Ghub community a single community place, accessible from anywhere in the world via a web browser, and hosts analytical tools, data, and other shared resources.
- Who develops the analytical tools on Ghub? You and others member of the Ghub community.
- Here we outline the procedure for developing a Pegasus WMS computational workflow analytical tool on Ghub.

- Computational workflows are a formalization of the manual computational workflow job steps that a scientist performs to obtain scientific results.
- A Workflow Management System (WMS) comprises software that manages the distribution and execution of computational workflows.
- The Pegasus WMS was selected as the best general purpose WMS to provide the structured platform required for implementing computational workflows on the HUBzero platform [1, 2, 3].

Example Pegasus WMS Architecture

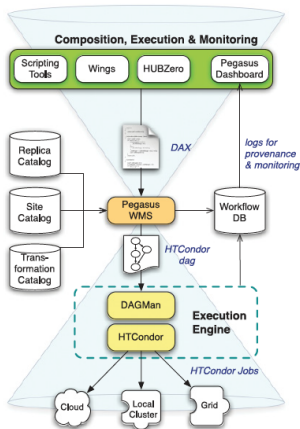


Figure 1: Pegasus WMS Architecture Example[4]

- HTCondor is a workload scheduling system for computational jobs.
- HTCondor provides a job queuing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management.
- DAGMan is a HTCondor tool that allows multiple jobs to be organized as a workflow, represented as a directed acyclic graph (DAG) in which the nodes represent computational tasks and edges represent the dependencies of those tasks [5].
- DAGMan automatically submits workflow jobs such that certain jobs need to be complete before other jobs start running.
- DAGMan provides the workflow engine for Pegasus.

- Pegasus requires three information catalogs to plan a workflow. These are the Site Catalog, the Transformation Catalog and the Replica Catalog.
- The Site Catalog describes the site where the workflow jobs are to be executed. For Ghub, workflow jobs are executed on CCR's UB-HPC cluster general-compute nodes by interfacing with CCR's SLURM (Simple Linux Utility for Resource Management) Workload Manager.
- The Transformation Catalog describes the launch scripts invoked by the workflow jobs. The launch scripts loads and then runs executable modules available on CCR. Execution of workflow jobs on UB-HPC requires access to executable modules available on CCR.
- The Replica Catalog tells Pegasus where to find the input files required by the workflow jobs.

- Pegasus comprises a set of system components which can be invoked via command line tools or an API interface.
- On Ghub, a Jupyter Notebook provides the interface which allows the user to interface with Pegasus, via Pegasus Python API commands, to create a YAML formatted file defining the workflow jobs, the input and output for the jobs, and the job dependencies.
- The Ghub submit command line tool allows Pegasus jobs to be executed remotely.
- The Ghub submit command line tool is used to plan and launch Pegasus workflow jobs via the Pegasus kickstart process.

- Pegasus processes the YAML file, resolving data and software locations and all required data movements, and creates the required information catalogs and compiles a DAG file. DAGMan, as directed by the DAG file, orders the jobs according to their dependencies, and submits the jobs ready for execution to HTCondor.
- SLURM provides the framework for queuing jobs, allocating compute nodes, and starting the execution of jobs. When a SLURM job execution completes, the final status of the finished job is returned to Pegasus.

Pegasus WMS Workflow Process Steps I

- Step 1: Get user's workflow parameters.
- Step 2: Create a workflow using Pegasus API commands.
- Step 3: For each executable required by the workflow:
 - Create a launch script for the executable.
 - Add the launch script to the Transformation Catalog.
- Step 4: For each parallel run of the launch script(s):
 - Create a job for the launch script.
 - Add arguments for the job based on user's parameters.
 - Add inputs and outputs for the job.
 - Add dependencies for the job.
 - Add the job to the workflow.
 - Add the inputs for the job to the Replica Catalog.

Pegasus WMS Workflow Process Steps II

Step 5: Create the YAML file for the workflow.

Step 6: Plan and submit the YAML file.

Step 7: Wait for the workflow to complete.

Step 8: View and analyze the workflow's output.

- The procedure to implement a simple pipeline Pegasus workflow is demonstrated with the [Ghub_Pegasus_WMS_MATLAB_Example](#) project.
- This project's workflow executes MATLAB scripts on CCR to perform coordinate conversions.
- Clone the [Ghub_Pegasus_WMS_MATLAB_Example](#) on Ghub and follow the steps outlined by the README.md file.

Simple Pipeline Workflow Architecture

Pegasus WMS Workflow MATLAB Example

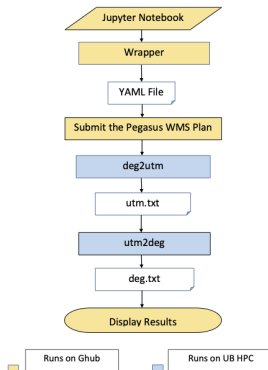


Figure 2: User Interface

Workflow's Output

```
Pegasus workflow in progress. This should take approximately 30 minutes...
self.parent:
self.tooldir: /home/thehub/renettej/AAA_notebooks/ghubex3
self.bindir: /home/thehub/renettej/AAA_notebooks/ghubex3/bin
self.datadir: /home/thehub/renettej/AAA_notebooks/ghubex3/data
self.workingdir: /home/thehub/renettej/AAA_notebooks/ghubex3
self.rundir:
self.latitude: 42.886447
self.longitude: -78.878369
self.maxwalltime: 10
Wrapper_5_0_1...
tooldir: /home/thehub/renettej/AAA_notebooks/ghubex3
matlab_launch_exec_path: /home/thehub/renettej/AAA_notebooks/ghubex3/remotebin/matlabLaunch.sh
Run 38987 registered 1 job instance. Fri Dec 1 12:46:02 2023
Run 38987 instance 1 released for submission. Fri Dec 1 12:46:32 2023
(413970.0) DAG Submitted at WF-ccr-ghub Fri Dec 1 12:46:36 2023
(413970.0) DAG Running at WF-ccr-ghub Fri Dec 1 12:47:51 2023
(413970.0) DAG Running at WF-ccr-ghub Fri Dec 1 12:53:27 2023
(413970.0) DAG Running at WF-ccr-ghub Fri Dec 1 12:59:27 2023
(413970.0) DAG Running at WF-ccr-ghub Fri Dec 1 13:06:07 2023
(413970.0) DAG Running at WF-ccr-ghub Fri Dec 1 13:11:27 2023
(413970.0) DAG Done at WF-ccr-ghub Fri Dec 1 13:19:28 2023

Workflow elapsed time: 33.518590517838796 [min]

Workflow completed successfully
```

Figure 3: Workflow Output

Workflow's UB CCR Scratch Directory

Remote site: /panfs/panfs.cbis.ccr.buffalo.edu/scratch/grp-ghub/ghubjobs/1701492362_00038987_01/scratch

1701477133_00038981_01
1701492362_00038987_01
scratch

Filename ^	Filesize	Filetype	Last modified
..			
._time_results.00038987_01	59	00038987..	12/01/2023 16:09:49
._timestamp_finish.00038987_01	11	00038987..	12/01/2023 16:09:49
._timestamp_start.00038987_01	11	00038987..	12/01/2023 16:09:11
._timestamp_transferred.00038987_01	11	00038987..	12/01/2023 16:09:09
00038987_01_413974.slurm	1,862	slurm-file	12/01/2023 15:47:20
00038987_01_413975.slurm	1,879	slurm-file	12/01/2023 16:06:31
00038987_01_413977.slurm	1,858	slurm-file	12/01/2023 16:08:45
chmod	58,584	File	11/16/2020 17:24:58
chmod_matlablaunch_ID0000001_0.stderr	0	stderr-file	12/01/2023 15:47:52
chmod_matlablaunch_ID0000001_0.stdout	3,792	stdout-file	12/01/2023 15:47:53
deg.txt	21	txt-file	12/01/2023 16:09:47
deg2utm	105,296	File	12/01/2023 15:47:05
matlablaunch	809	File	12/01/2023 15:47:06
matlablaunch_ID0000001.stderr	0	stderr-file	12/01/2023 16:06:58
matlablaunch_ID0000001.stdout	4,729	stdout-file	12/01/2023 16:07:45
matlablaunch_ID0000002.stderr	0	stderr-file	12/01/2023 16:09:11
matlablaunch_ID0000002.stdout	4,698	stdout-file	12/01/2023 16:09:49
slurm_00038987_01_413974.stderr	0	stderr-file	12/01/2023 15:47:52
slurm_00038987_01_413974.stdout	0	stdout-file	12/01/2023 15:47:52
slurm_00038987_01_413975.stderr	0	stderr-file	12/01/2023 16:06:58
slurm_00038987_01_413975.stdout	0	stdout-file	12/01/2023 16:06:56
slurm_00038987_01_413977.stderr	0	stderr-file	12/01/2023 16:09:10
slurm_00038987_01_413977.stdout	0	stdout-file	12/01/2023 16:09:10
utm.txt	25	txt-file	12/01/2023 16:07:44
utm2deg	105,423	File	12/01/2023 15:47:11

Figure 4: UB CCR Scratch Directory



Michael McLennan, Steven Clark, Ewa Deelman, Mats Rynge, Karan Vahi, Frank McKenna, and Derrick Kearney. Bringing Scientific Workflow to the Masses via Pegasus and HUBzero. *SemanticScholar*, 2013.



Michael McLennan, Steven Clark, Ewa Deelman, Mats Rynge, Karan Vahi, Frank McKenna, Derrick Kearney, and Carol Song. HUBzero and Pegasus: integrating scientific workflows into science gateways. *Wiley*, 2014.



Ewa Deelman, Karan Vahi, Gideon Juve, Mats Rynge, Scott Callaghan, Philip J. Maechling, Rajiv Mayani, Weiwei Chen, Rafael Ferreira da Silva, Miron Livny, and Kent Wenger. Pegasus, a workflow management system for science automation. *Future Generation Computer Systems*, 46:17–35, 2015. ISSN: 0167-739X. DOI: <https://doi.org/10.1016/j.future.2014.10.008>. URL: <https://www.sciencedirect.com/science/article/pii/S0167739X14002015>.



C. S. Liew, M. P. Atkinson, M. Galea, T. F. Ang, P. Martin, and J. I. V. Hemert. Scientific workflows. *Computing Surveys*, 49(4):1–39, 2016. DOI: <https://doi.org/10.1145/3012429>.



Ryan Mitchell, Loïc Pottier, Steve Jacobs, Rafael Ferreira da Silva, Mats Rynge, Karan Vahi, and Ewa Deelman. Exploration of Workflow Management Systems Emerging Features from Users Perspectives. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 4537–4544, 2019. DOI: 10.1109/BigData47090.2019.9005494.