



Pegasus Users Group

MEETING



Pegasus is funded by the National Science Foundation under grant #1664162



Pegasus 5.0: Features

Karan Vahi

Senior Computer Scientist

Date: Feb 23rd, 2021

Pegasus 5.0

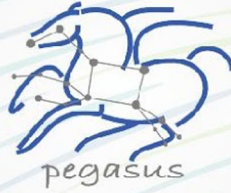


Released Nov, 2020

- New and fresh Python3 API to compose, submit and monitor workflows, and configure catalogs
 - Developed brand new from grounds up.
 - New **YAML** based format to describe workflows
 - Allows for creation of each of the catalogs (site, transformation, replica and properties)
 - Allows you to plan/submit/monitor/analyze/statistics of your workflow
- Python 3 support
 - All Pegasus tools are Python 3 compliant.
 - 5.0 release requires Python 3 on workflow submit node
 - Python PIP packages for workflow composition and monitoring
 - <https://pypi.org/project/pegasus-wms/>

Tip: JAVA and R API's now emit workflow descriptions in YAML like the new python API.

Pegasus 5.0



- Adoption of YAML

- Abstract Workflow
- Replica Catalog
- Transformation Catalog
- Site Catalog
- Kickstart Provenance Records



Converters provided to convert existing catalogs

- *pegasus-sc-converter* , *pegasus-rc-coverter*, *pegasus-tc-converter*

- Reworked Documentation

- *readthedocs* style documentation using *restructured* text
- Broken into a *user* guide and a *reference* guide



Read the Docs

- Brand new *Jupyter* Notebook based Tutorial

- Check out the tutorial later today!



Pegasus 5.0



- Zero configuration required to submit to local HTCondor pool.
 - The “*hello world*” example on the right will work out of the box
 - Pegasus will automatically create sensible defaults for sites
 - local
 - condorpool
 - By default, site “*condorpool*” is used as execution site.
 - Site “*local*” still designates the submit node, and is used to run Pegasus auxiliary jobs.

```
#!/usr/bin/env python3
import logging
import sys

from Pegasus.api import *

# logs to be sent to stdout
logging.basicConfig(level=logging.DEBUG, stream=sys.stdout)

# --- Transformations -----
echo = Transformation(
    "echo",
    pfn="/bin/echo",
    site="condorpool"
)

tc = TransformationCatalog()\
    .add_transformations(echo)

# --- Workflow -----
Workflow("hello-world", infer_dependencies=True)\
    .add_jobs(
        Job(echo)
        .add_args("Hello World")
        .set_stdout("hello.out")
    ).add_transformation_catalog(tc)\
    .plan(submit=True)\
    .wait()
```


Pegasus 5.0

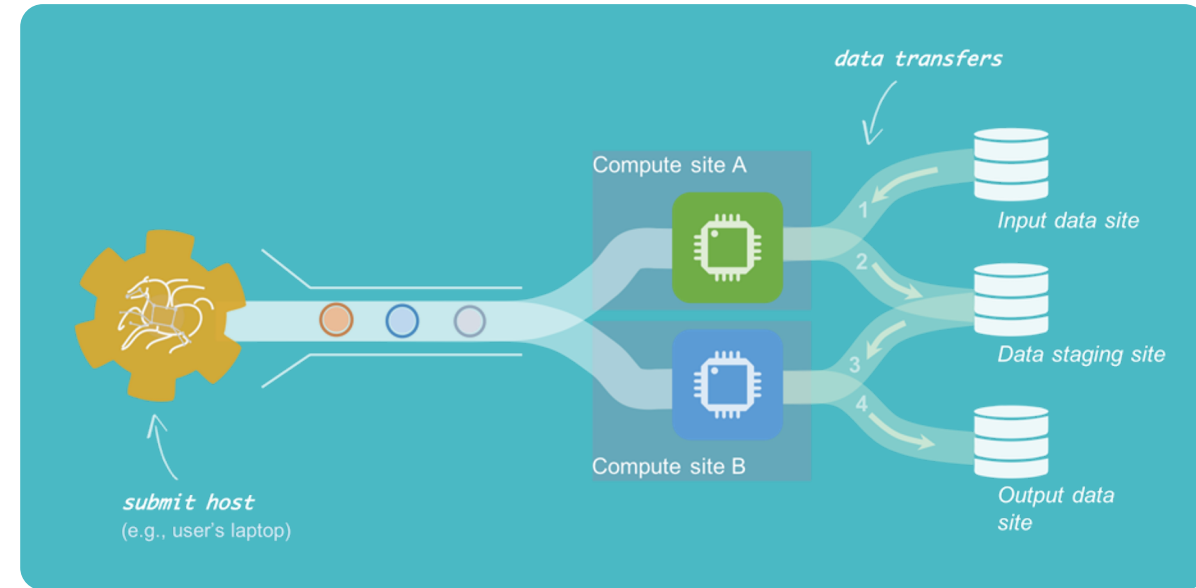


- **Default Data Configuration**

- IS “**condorio**” from “**sharedfs**” earlier.
- Worker nodes do not share a file system
- Data is pulled from / pushed to the submit host via HTCondor file transfers
- Staging site is the submit host

- **Existing users if operating in sharedfs mode now need to set**

- **pegasus.data.configuration = sharedfs** in their properties



Pegasus 5.0



Input Replica Catalog

Discover the location of input files or previously generated datasets to use for planning purposes.

- **Configuration:** use the properties prefix *pegasus.catalog.replica*

Output Replica Catalog

- Registers outputs including file metadata such as size and checksums
- By default Pegasus will registers outputs to a JDBC based Replica Catalog (*workflow-name.replicas.db*) in the workflow submit directory.
- For hierarchical workflows only one output replica catalog db is generated in the root workflow submit directory.
- **Configuration:** use the properties prefix *pegasus.catalog.replica.output*
- **Data Reuse:** use *—reuse* option to pegasus-plan to pass submit directories of previous runs

Note: In 4.9.x and before, the input replica catalog was used for registration of outputs.

Pegasus 5.0



Hierarchical Workflow Improvements

Automatic handing of data dependencies between pegasusWorkflow (dax) jobs and compute jobs

```
x-pegasus: {apiLang: python, createdBy: bamboo, createdOn: '07-10-20 11:09:29'}
pegasus: '5.0'
name: local-hierarchy
jobs:
- type: pegasusWorkflow
  file: blackdiamond.yml
  id: ID0000001
  arguments: [--input-dir, input, --output-sites, local, -vvv, --force]
  uses:
  - {lfn: blackdiamond.yml, type: input}
  - {lfn: f.d, type: output, stageOut: true, registerReplica: true}
- type: job
  namespace: diamond
  version: '4.0'
  name: post-analyze
  id: ID0000002
  arguments: [-a, post-analyze, -T, '60', -i, f.d, -o, f.e]
  uses:
  - {lfn: f.d, type: input}
  - {lfn: f.e, type: output, stageOut: true, registerReplica: true}
jobDependencies:
- id: ID0000001
  children: [ID0000002]
```

pegasusWorkflow job (ID0000001) generates an output file “f.d”

Compute job (ID0000002) in the same workflow can use “f.d” as input

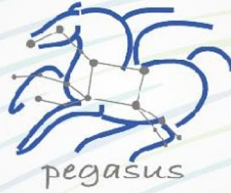
Pegasus 5.0



Data Management Improvements

- Ability to do bypass staging of files at a per file, executable and container level
 - Set the *"bypass"* flag for file/executable/container
 - Useful when you want to pull the container down only once from Docker|Singularity Hub, but do bypass for other input data
- Support for integrity checking of user executables and application containers in addition to data
- WebDAV support
- Support for staging of generated outputs to multiple output sites

Pegasus 5.0



New Common Credential File

- Manage simple credentials such as username / password / accesskey / secretkey / tokens in one place
- WebDAV, S3 - more to come
- ~/.pegasus/credentials.conf

Credentials Pre-flight Check

- Planner now does simple existence and permission checks for local credentials.

```
# For simple username/password protocols, such as WebDAV,  
# just specify the hostname and credentials. In this  
# example, the credentials would be used for URLs  
# matching the section, such as  
# webdav://data.cyverse.org/some/file.txt
```

```
[data.cyverse.org]
```

```
username = joe  
password = secretsauce1
```

```
# For S3 access, you can create an entry for the cloud  
# specific options, and then one or more user specific  
# entries with a key @ matching the cloud one (for  
# example, [amazon] and [joe@amazon] below)
```

```
[amazon]  
endpoint = https://s3.amazonaws.com/
```

```
[joe@amazon]  
access_key = 90c4143642cb097c88fe2ec66ce4ad4e  
secret_key = ababababababababababababababababababab
```


Pegasus 5.0



Monitoring Improvements

- Unicode compatibility for databases. We also enforce a consistent UTF-8 environment
- We record “maxrss” the maximum physical memory used by a job during it’s execution
- average cpu utilization ($\text{utime} + \text{stime} / \text{duration}$)
 - Utime: CPU time spent in user code
 - Stime: CPU time spent in kernel code
 - Duration: total runtime of a job
- Pegasus-statistics will report these metrics
- **Useful for creating application profiles.**

Pegasus 5.0



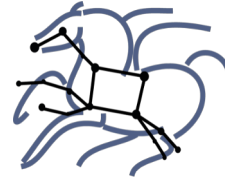
CWL Support: pegasus-cwl-converter

- Convert a subset of CWL v1.1 to native Pegasus yaml format
- Automatic generation of Pegasus workflow, replica, and transformation catalogs into single file



COMMON
WORKFLOW
LANGUAGE

```
my_cwl_workflow
├── tar.cwl
├── compile_1.cwl
├── compile_2.cwl
├── get_file_sizes.cwl
├── get_file_sizes.sh
├── src_tarball
├── input.yml
└── workflow.cwl
```



```
apiLang: python
createdBy: ryantanaka
createdOn: 07-24-20T10:08:48Z
pegasus: "5.0"
name: compile
replicaCatalog:
  replicas: [...]
transformationCatalog:
  transformations: [...]
jobs: [...]
jobDependencies: [...]
```



Thank You!