Pegasus 5.0: Features

Karan Vahi
Senior Computer Scientist

Date: Feb 23rd, 2021
Pegasus 5.0

- 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/](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

• 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.

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

from Pegasus.api import *

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 register 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.
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:
  - {fn: blackdiamond.yml, type: input}
  - {fn: 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:
  - {fn: f.d, type: input}
  - {fn: 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
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
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.
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 (utime+stime/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

Manpage: https://pegasus.isi.edu/docs/5.0.0dev/manpages/pegasus-cwl-converter.html
Thank You!