Pegasus 101

Rafael Ferreira da Silva

rafsilva@isi.edu

February 25, 2021

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

Why Pegasus?

Automates Complex, Multi-stage Processing Pipelines

Enables Parallel, **Distributed Computations**

Automatically Executes Data Transfers

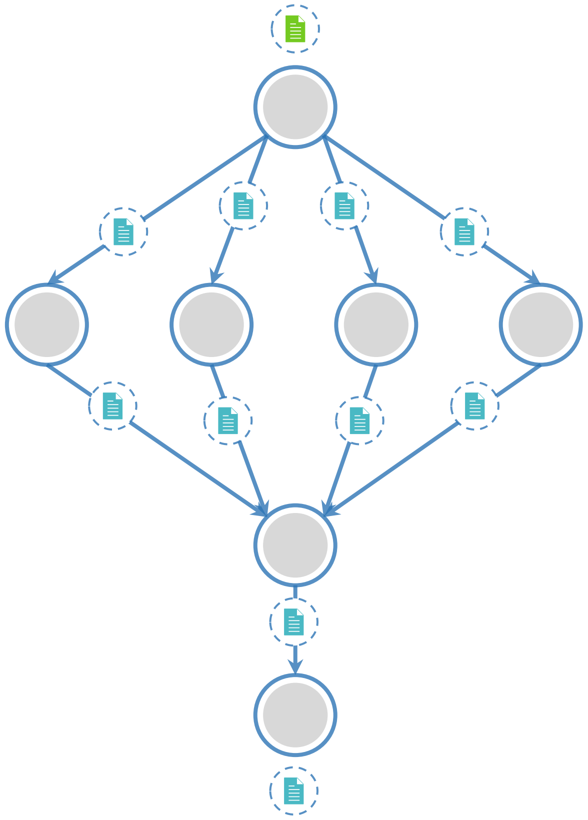
Reusable, Aids **Reproducibility**

Records How Data was Produced **(Provenance)**

Handles **Failures** with to Provide Reliability

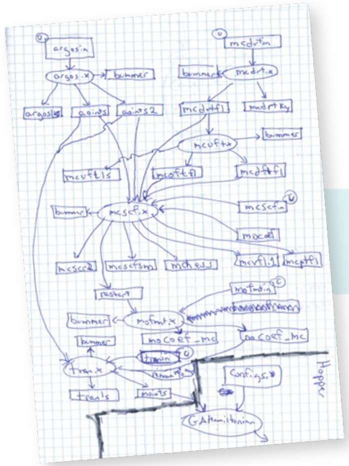
Keeps Track of Data and **Files**

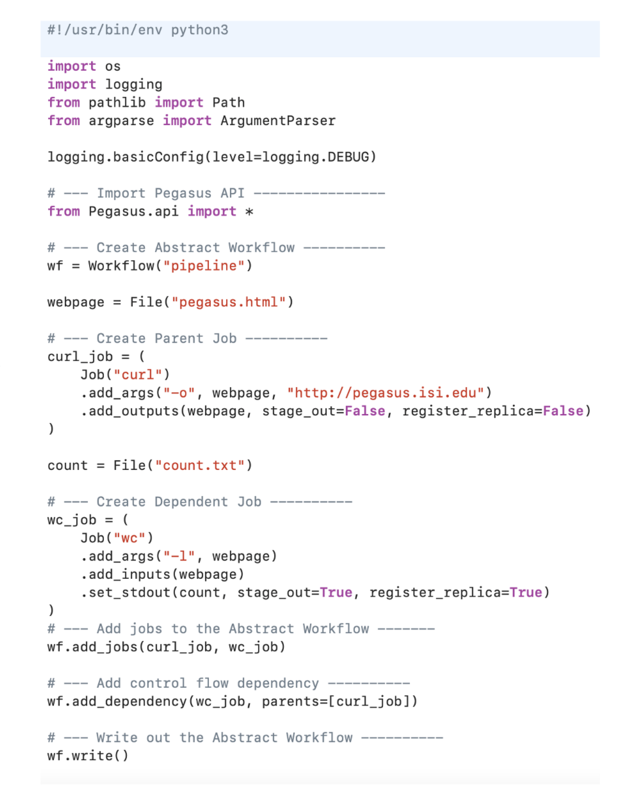
Ensures **Data Integrity** during workflow execution



NSF funded project since 2001,
with close collaboration with HTCondor team

How to build workflows with Pegasus?





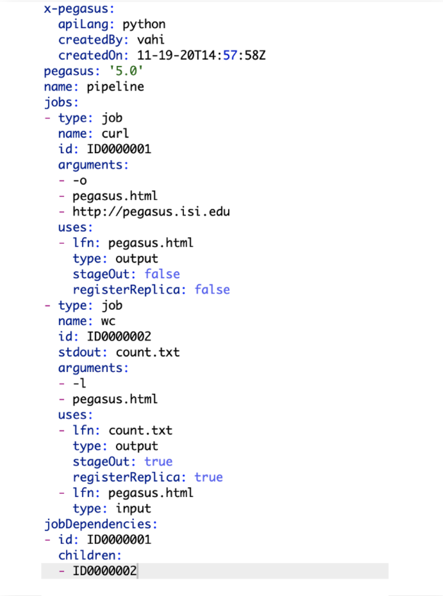
```
#!/usr/bin/env python3

import os
import logging
from pathlib import Path
from argparse import ArgumentParser

logging.basicConfig(level=logging.DEBUG)

# --- Import Pegasus API ----------------
from Pegasus.api import *

# --- Create Abstract Workflow ----------
wf = Workflow("pipeline")

webpage = File("pegasus.html")

# --- Create Parent Job ----------
curl_job = (
    Job("curl")
    .add_args("-o", webpage, "http://pegasus.isi.edu")
    .add_outputs(webpage, stage_out=False, register_replica=False)
)

count = File("count.txt")

# --- Create Dependent Job ----------
wc_job = (
    Job("wc")
    .add_args("-l", webpage)
    .add_inputs(webpage)
    .set_stdout(count, stage_out=True, register_replica=True)
)
# --- Add jobs to the Abstract Workflow -------
wf.add_jobs(curl_job, wc_job)

# --- Add control flow dependency ----------
wf.add_dependency(wc_job, parents=[curl_job])

# --- Write out the Abstract Workflow ----------
wf.write()
```

python™

jupyter

Java

R



```
x-pegasus:
  apiLang: python
  createdBy: vahi
  createdOn: 11-19-20T14:57:58Z
pegasus: '5.0'
name: pipeline
jobs:
- type: job
  name: curl
  id: ID0000001
  arguments:
  - -o
  - pegasus.html
  - http://pegasus.isi.edu
  uses:
  - lfn: pegasus.html
    type: output
    stageOut: false
    registerReplica: false
- type: job
  name: wc
  id: ID0000002
  stdout: count.txt
  arguments:
  - -l
  - pegasus.html
  uses:
  - lfn: count.txt
    type: output
    stageOut: true
    registerReplica: true
  - lfn: pegasus.html
    type: input
jobDependencies:
- id: ID0000001
  children:
  - ID0000002
```

*Try our **self-guided tutorial** available in the Pegasus website!*

What information does Pegasus need?

from the abstraction to execution

Site Catalog

Describes the **execution sites** where the workflow jobs are to be executed

***automatically created for default local and condorpool sites*

Transformation Catalog

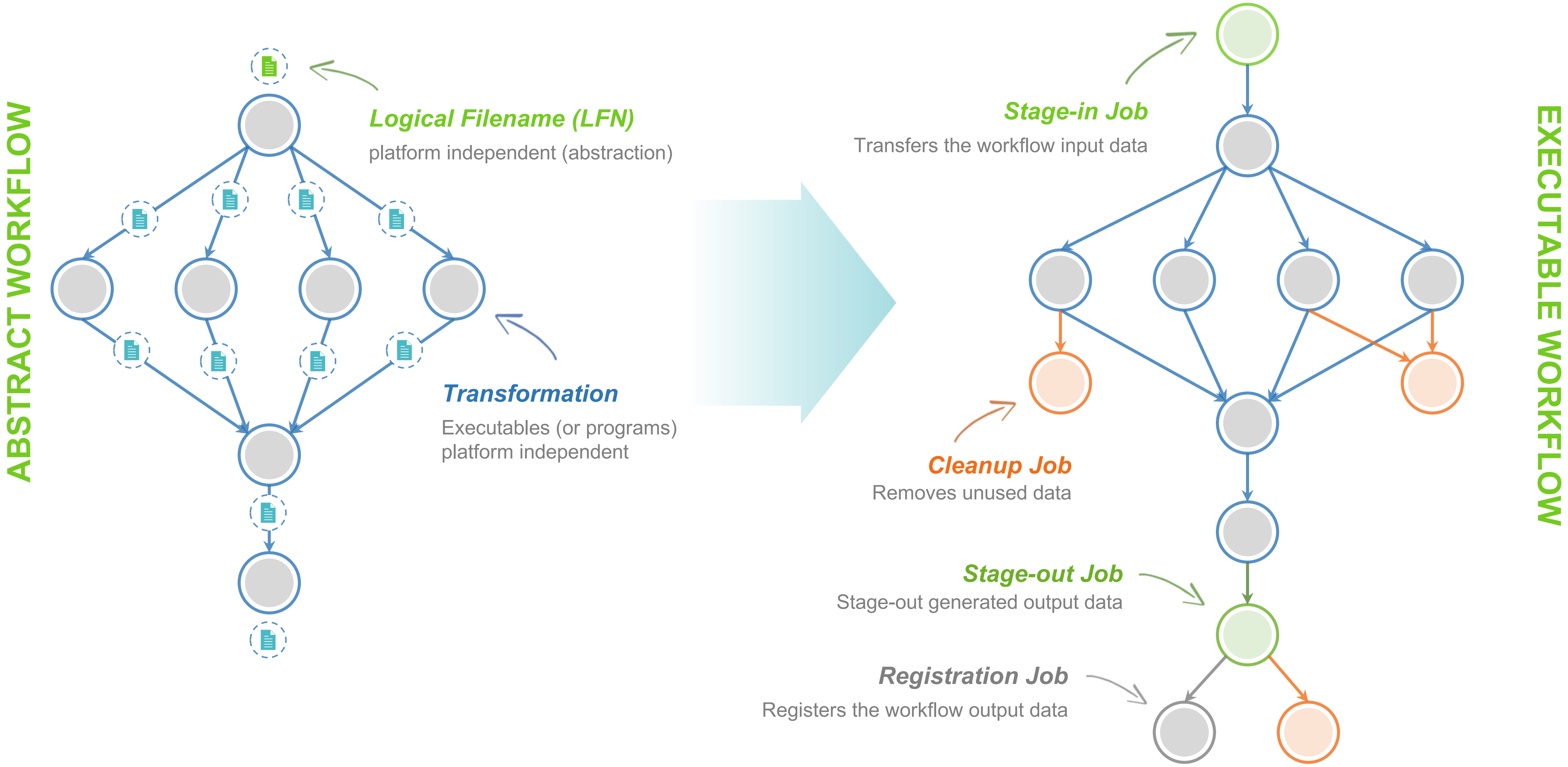
Describes the **executables** (called “transformations”) used by the workflow

Replica Catalog

Describes all of the **input data** stored on external servers

Portable Description

Users do not worry about low level execution details



ABSTRACT WORKFLOW

Logical Filename (LFN)

platform independent (abstraction)

Transformation

Executables (or programs)
platform independent

Stage-in Job

Transfers the workflow input data

Cleanup Job

Removes unused data

Stage-out Job

Stage-out generated output data

Registration Job

Registers the workflow output data

EXECUTABLE WORKFLOW

Could you talk more about execution?

COMPUTE

Desktop/Laptop

Local/Campus Cluster

HTCondor, PBS, Slurm, LSF, SGE

HPC Systems

XSEDE, TACC, ORNL, ANL, NERSC, etc.

Clouds

Amazon AWS, Google Cloud, Chameleon Cloud, etc.

Grids

Open Science Grid

STORAGE

Transfer Protocols

HTTP, SCP, GridFTP, Globus Online, iRods, Amazon S3, Google Storage, SRM, FDT, Stashcp, Rucio, cp, ln -s

File Systems

Shared and non-shared file systems, and HTCondor I/O

Parallel Transfers

Automated Retries

OPTIMIZATIONS

Task Clustering

Reduces execution overhead

Data Reuse

Avoids re-computations

Fault-tolerance

Checkpoints, Retries, Rescue DAGs

Large-scale Workflows

Hierarchical execution

What to do next?

Grab us during the **break**

Come to **office hours** @12:30pm PST / 3:30pm EST

Do a **self-guided tutorial**

https://pegasus.isi.edu/documentation/user-guide/tutorial.html

We are happy to learn about your application and are here to help