Pegasus workflows on the Open Science Grid infrastructure

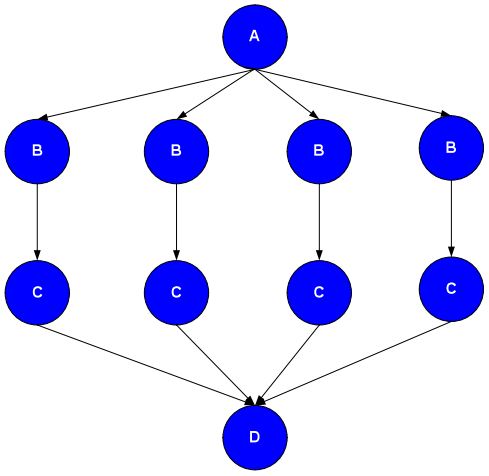
OSG Campus Infrastructures Community iForum 8/22/13

Mats Rynge

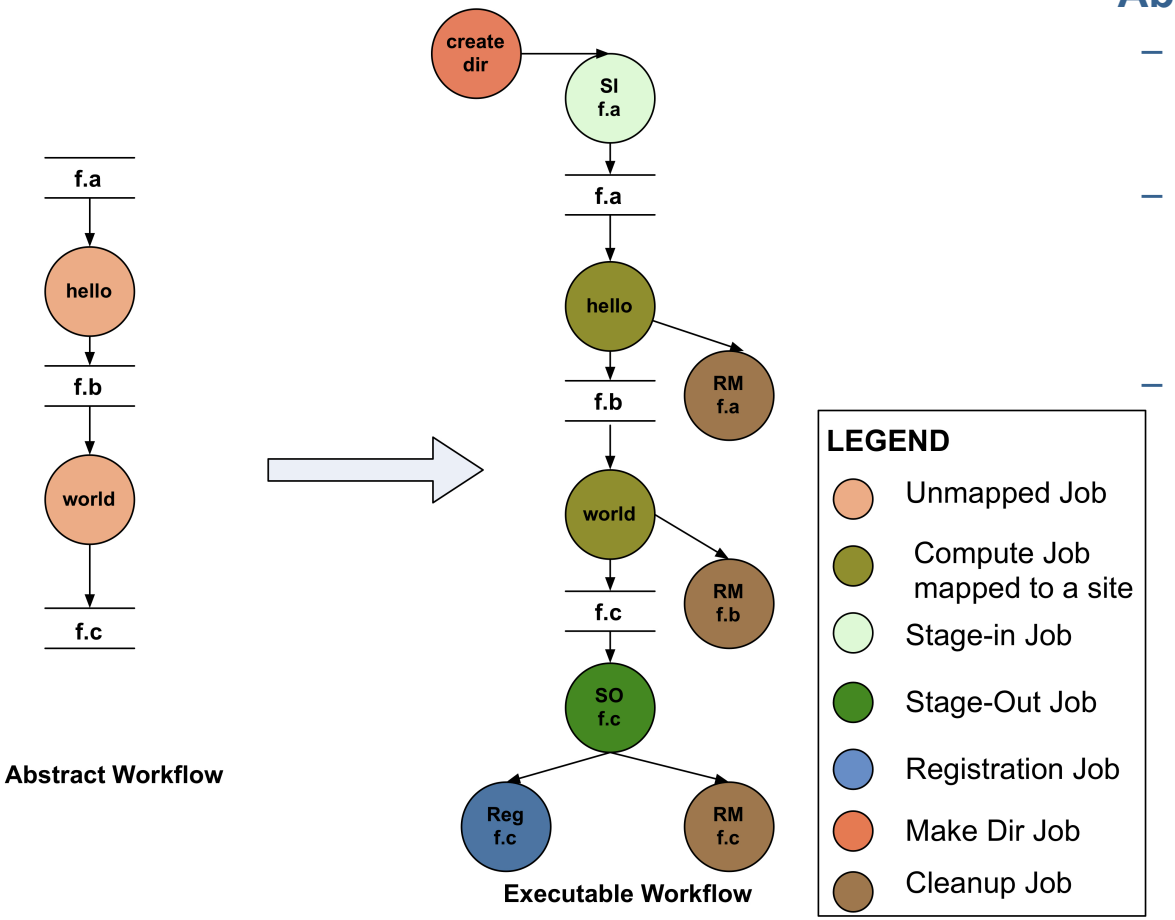
USC Information Sciences Institute

Pegasus Workflow Management System

- Builds on top of Condor DAGMan.
- Abstract Workflows - Pegasus input workflow description
  - Workflow “high-level language”
  - Only identifies the computation, devoid of resource descriptions, devoid of data locations
- Pegasus is a workflow planner/mapper (“compiler”)
  - Target is DAGMan DAGs and Condor submit files
  - Transforms the workflow for performance and reliability
  - Automatically locates physical locations for both workflow components and data
  - Collects runtime provenance



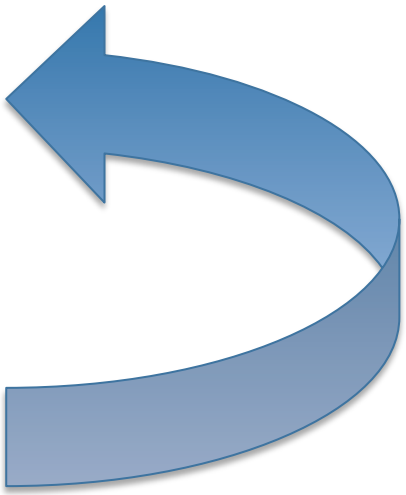
Abstract to Executable Workflow Mapping



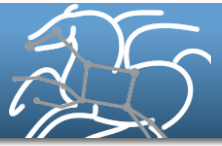
- Abstraction provides
  - Ease of Use (do not need to worry about low-level execution details)
  - Portability (can use the same workflow description to run on a number of resources and/or across them)
  - Gives opportunities for optimization and fault tolerance
    - automatically restructure the workflow
    - automatically provide fault recovery (retry, choose different resource)

Workflow components: DAX and Catalogs

- **Abstract workflow description (DAX)**
  - Defines the compute tasks and dependencies (directed acyclic graph)
- **Transformation catalog**
  - Defines executables used by the workflow
  - Executables can be installed in different locations at different sites
- **Replica catalog**
  - Locations of existing data products – input files and intermediate files from previous runs
- **Site catalog**
  - Defines the execution environment and potential data staging resources
  - Simple in the case of Condor pool, but can be more complex when running on grid resources



Generated at submit time

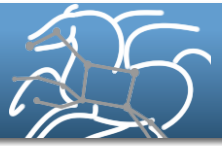


Workflow Monitoring

- **Monitoring framework with DB backend**
  - Populates data at runtime. A background daemon monitors the logs files and populates information about the workflow to a database
  - Stores workflow structure, and runtime stats for each task.

- **Tools for querying the monitoring framework**
  - pegasus-status
    - Status of the workflow
  - pegasus-statistics
    - Detailed statistics about your finished workflow
  - pegasus-plots
    - Visualization of your workflow execution

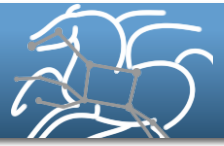
```
------------------------------------------------------------------------------
Type           Succeeded Failed  Incomplete  Total     Retries   Total+Retries
Tasks          135002    0       0           135002    0         135002
Jobs           4529      0       0           4529      0         4529
Sub-Workflows  2         0       0           2         0         2
------------------------------------------------------------------------------

Workflow wall time                                : 13 hrs, 2 mins, (46973 secs)
Workflow cumulative job wall time                 : 384 days, 5 hrs, (33195705 secs)
Cumulative job walltime as seen from submit side : 384 days, 18 hrs, (33243709 secs)
```

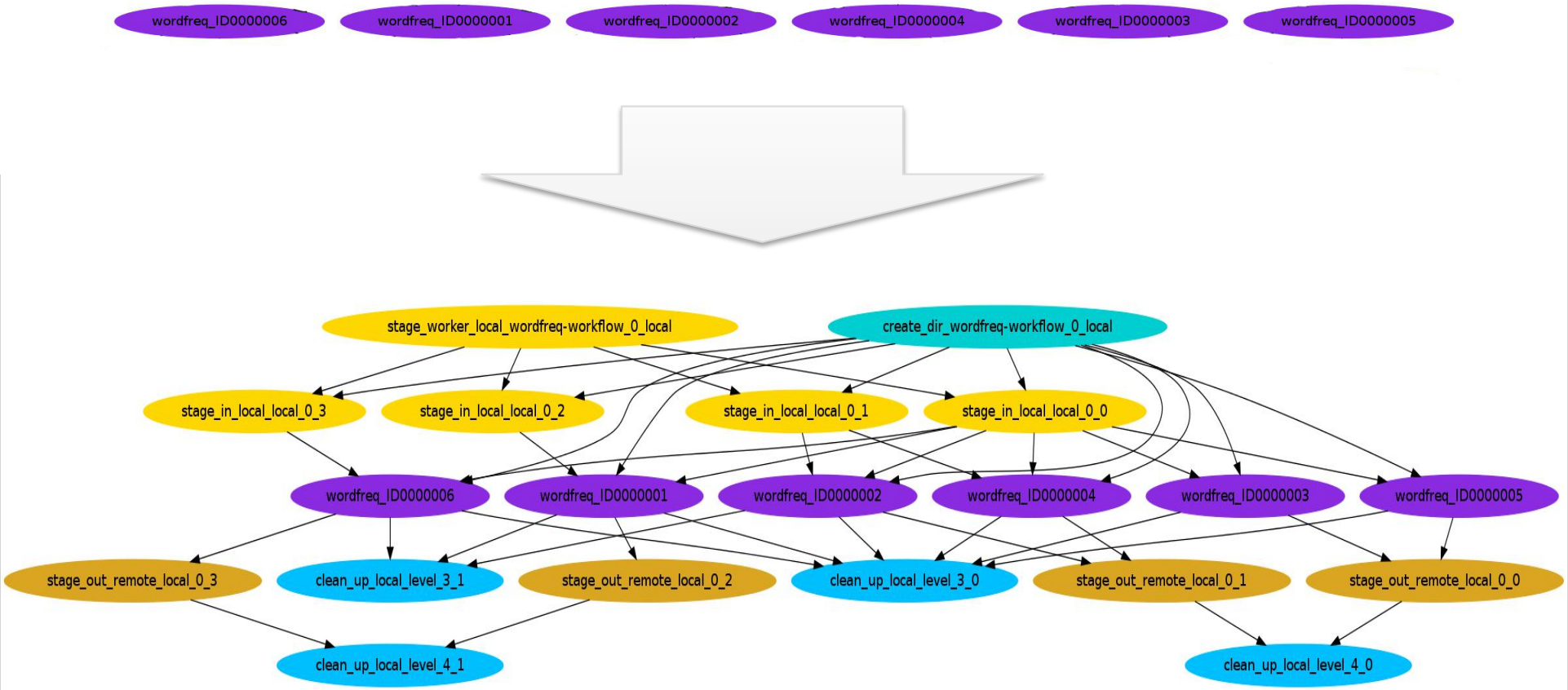
Workflow Debugging Through Pegasus

- After a workflow has completed, we can run pegasus-analyzer to analyze the workflow and provide a summary of the run

- pegasus-analyzer's output contains
  - a brief summary section
    - showing how many jobs have succeeded
    - and how many have failed.
  - For each failed job
    - showing its last known state
    - exitcode
    - working directory
    - the location of its submit, output, and error files.
    - any stdout and stderr from the job.



Simple workflow for the walkthrough



Relevant Links

- Pegasus: http://pegasus.isi.edu

- Tutorial and documentation: http://pegasus.isi.edu/wms/docs/latest/