Pegasus and Open Science Grid - a perfect match!

Mats Rynge

In true HTCondor fashion - the perfect match is both ways...

The Open Science Grid is a great execution environment for Pegasus workflows. OSG is based on HTCondor which makes it a vary "native" home for Pegasus.

The Open Science Grid is a distributed execution environment, with one of the challenges being how you get data to and from compute nodes. Pegasus has built in data management.
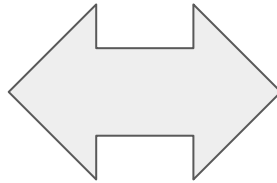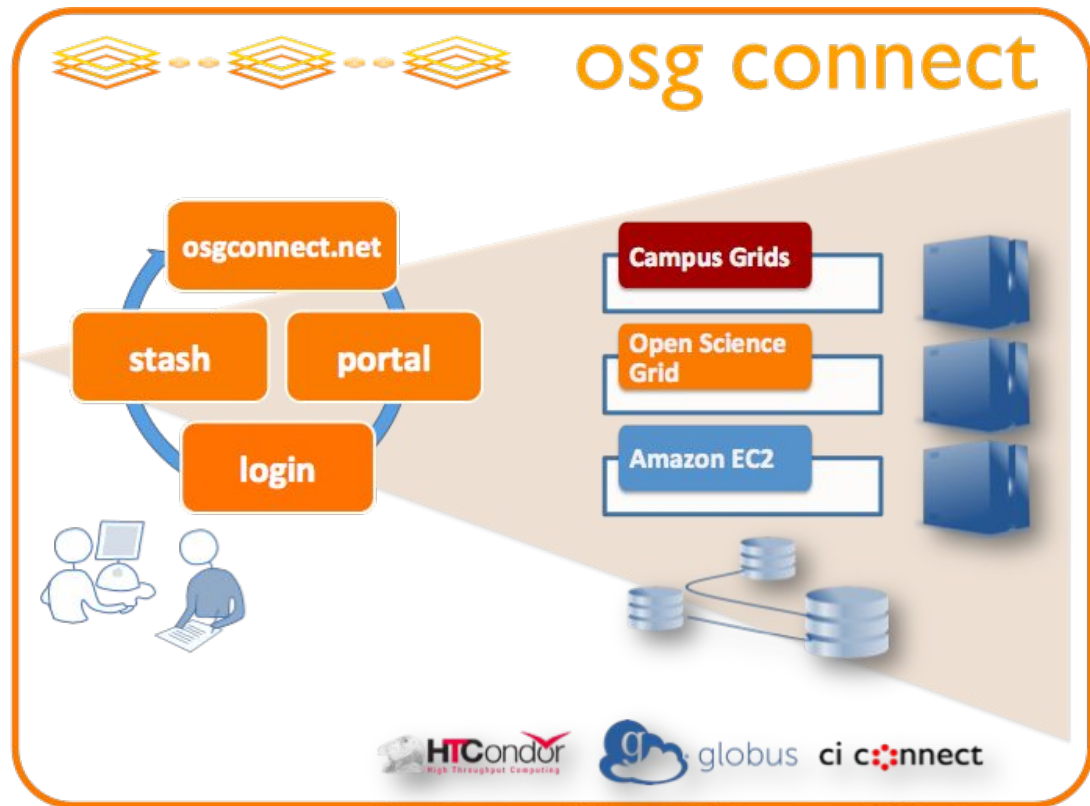
OSG Connect Service

Hosted submit hosts and data infrastructure

★ Login hosts
★ HTCondor
★ Storage
★ Software

Pegasus Deployment

Workflow Submit Node
- Pegasus WMS
- HTCondor

One or more Compute Sites
- Compute Clusters
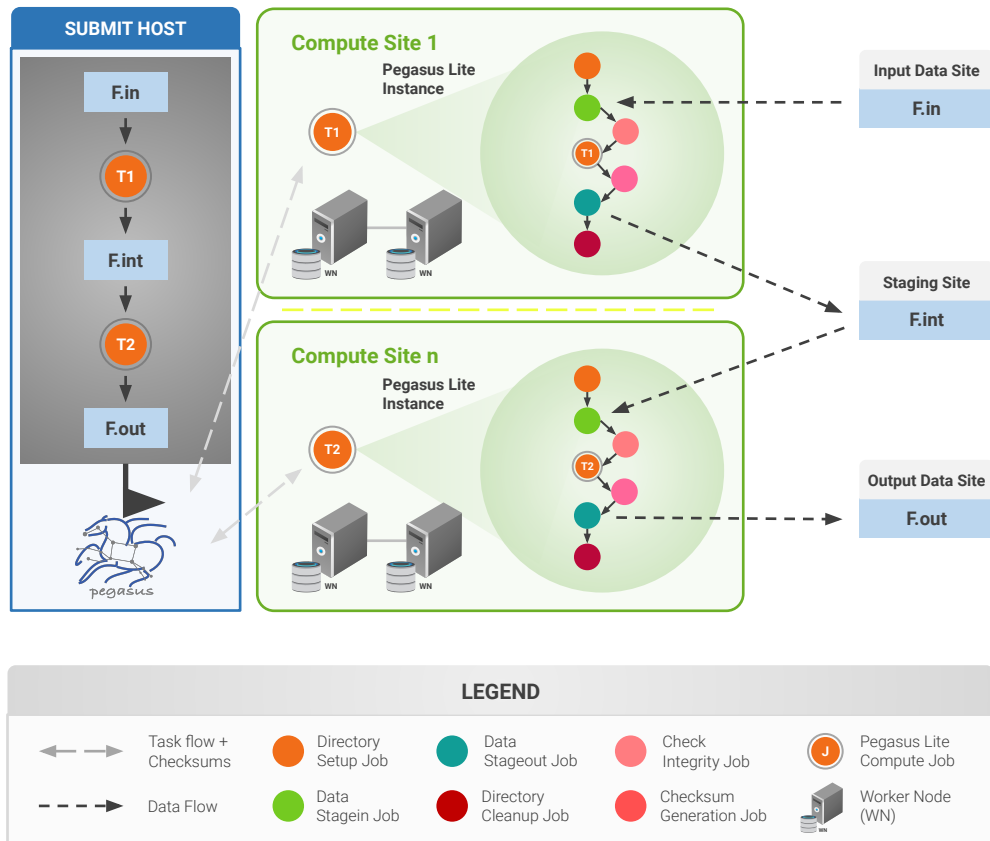- Cloud
- OSG

Input Sites
- Host Input Data

Data Staging Site
- Coordinate data movement for workflow

Output Site
- Where output data is placed

Data transfers - StashCache / WebDAV / S3 / GridFTP / …

- **CondorIO** - Well supported in Pegasus - no need for credentials

- **StashCache** - heavily cached method for getting data from OSGConnect to jobs. Can also be used for data from job to OSGConnect with SciToken auth (credential is applied to all OSGConnect jobs, and fully supported by Pegasus)

- **WebDAV** - Used by projects like Event Horizon Telescope for accessing data at Cyverse (could also use iRods). Third party copy extension to WebDAV called HTTP-TCP is being worked into OSG.

- **S3** - Limited support, but used in the past by for example Veritas.

- **GridFTP** - Being phased out due to X.509 GSI being phased out, but still widely used. Example: XENONnT for intermediate data, but RUCIO for inputs/outputs. Pegasus does currently not support RUCIO natively.
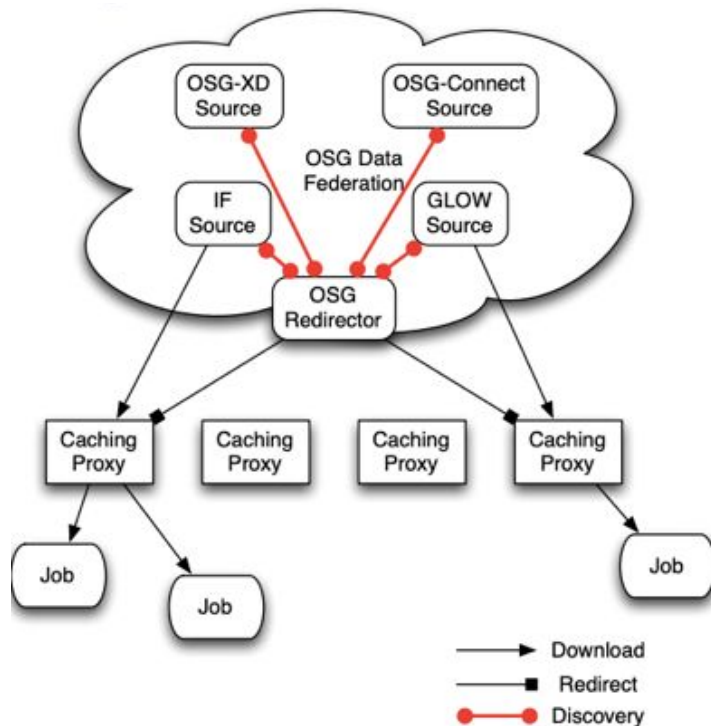
Introduction to Stashcache

- Caching infrastructure based on SLAC XRootD server & XRootD protocol.
- Cache servers are placed at several strategic cache locations across the OSG.
- Jobs utilize GeoIP to determine the nearest cache
- Job talks to the cache using HTTP(S) via CVMFS

Powered by:

XRootD

StashCache infrastructure (US)

Automatic Integrity Checking in Pegasus

Pegasus performs integrity checksums on input files right before a job starts on the remote node.

▶ For raw inputs, **checksums specified in the input replica catalog** along with file locations

▶ All **intermediate** and **output** files checksums are generated and tracked within the system.

▶ Support for **sha256** checksums

Job failure is triggered if checksums fail
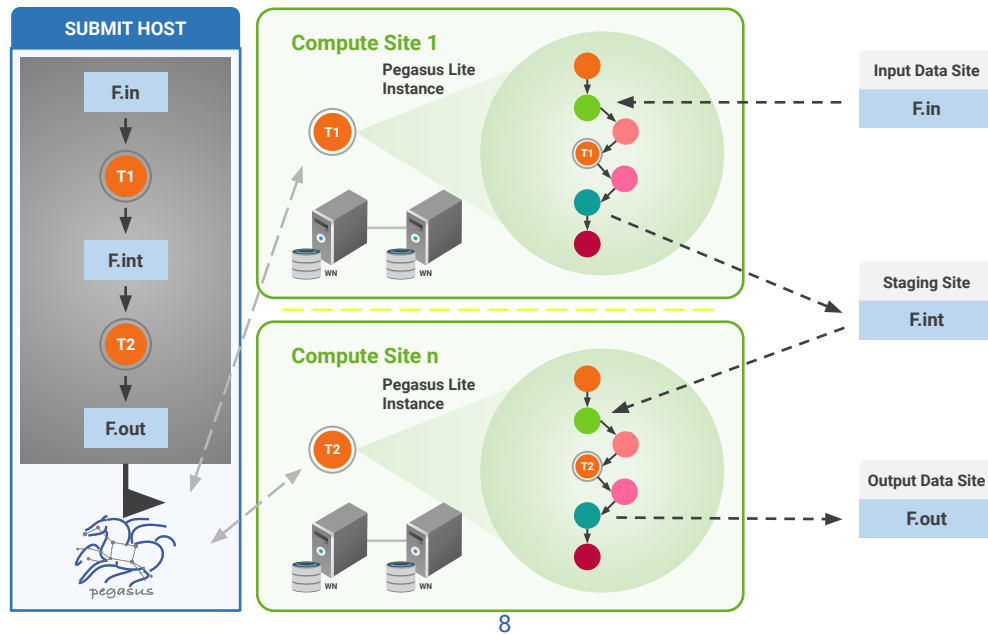
Containers on OSG

Containers in OSG are distributed and invoked by the infrastructure. Do **not** use Pegasus container functionality

```
Requirements = HAS_SINGULARITY == TRUE
+SingularityImage = "/cvmfs/singularity.opensciencegrid.org/opensciencegrid/osgvo-el7:latest"
```

Containers have to be registered first. More information:

https://support.opensciencegrid.org/support/solutions/articles/12000024676-docker-and-singularity-containers

Thank you!