Chapter 1. Conventions for Running Pegasus on the Grid

This document describes the conventions for workflow execution and job management needed to run workflows across multiple Grid sites.

1. Data Files

We first need to define how file names are referenced and processed. Pegasus is concerned with three kinds of file names:

LFN: the logical filenames used in the DAX. LFNs may be relative or absolute, and may or may not have multiple directory levels in them. When a LFN is looked up in a replica catalog (RC), it is looked up in its relative or absolute form, exactly as it is coded in the DAX file. Similarly, when an instance of an LFN is produced at a Grid site, it is stored in the RC exactly as it was coded in the DAX.

SFN: the physical "storage" filenames, as files are referenced by user jobs running at a grid site

TFN: the physical "transfer" filenames (i.e., complete URLs), as files are accessed by the GridFTP protocol, both inside and outside_ the site.

LFNs and SFNs are issues that the user needs to be concerned about. For the most part, the formation and use of the TFN happens automatically in the code generated by Pegasus. The user does not need to be aware of this coding of file names, other than to make sure that the GridFTP server is properly configured and identified to the site catalog. The important concept to understand here is how these names are used as a workflow is planned and executed on the Grid. File names are translated to SFNs by the workflow executor, and used by the programs that are executed in the workflow.

Input files are looked up in the Replica Catalog and copied to the site where the job will run, if they don't already exist at that site. In Pegasus v 2.0 RLS (http://www.globus.org/rls/), DB backend via JDBC and a Simple File are the available implementations of a replica catalog.

Programs start in their working directory (i.e., their "current working directory" (CWD) is set to the working directory before the job is started). When jobs execute, they reference their input files using names that have been translated into physical "storage file names" or SFNs. They create new output files at or below this current working directory. When jobs complete, any files they create that were designated in the DAX as "output" files are cataloged as existing at the site. In some cases, these files may be moved within the site for persistence. We anticipate that some jobs will need to have a "bigger" workspace, and may create files above or below their CWD . It is possible, but rare, that jobs will exist that cannot be accommodated by this model. Sometimes it may be necessary to adjust a job's parameters in order to get it to conform to this model. For programs that expect input files:

1. files should be pre-cataloged in the Replica Catalog by the logical name (LFN) exactly as coded in the DAX

2. the PFN in the Replica Catalog must be a TFN that can be used to retrieve the file.

3. the file is copied by Pegasus-inserted code to the storage directory(sdir), under the name relative to the LFN (regardless of whether the LFN is relative or absolute)

4. the SFN produced by Pegasus and placed in the command line is the file's relative name within the sdir.

2. Layout of a Grid Site

2.1. Grids Sites Properties

A Grid site, typically represented in a site monitoring entity , consists of the following storage areas:

- ***$DATA***: where users can create and leave persistent files, typically under VO-specific directory hierarchies.
- **$APP**: where users can create and leave application files and directories, typically under VO-specific directory hierarchies.
- **$TMP**: place to create temporary directories for the duration of a job or workflow. Typically shared by all users of the Grid site, and reachable and sharable by all worker nodes.
- **$WN_TMP**: place to create temporary directories on the local disk of the worker node. The local disk provides speed-up to IO intensive file operations, e.g. database searches.
- **$GRID:** location of the OSG (or VDT) software stack, under which we can expect the base directory for the Globus and Pegasus worker tools installation.

2.2. Description of Grid Sites in Pegasus

Pegasus defines the following concepts:

Each site has a grid storage directory ("**sdir**") that contains files that persist at the site:

- all files in the **sdir** are tracked in the Replica Catalog. The files that are transferred as part of the workflow description to the **sdir** are automatically registered in the Replica Catalog. The user can however explicitly ask them not to be registered using the transience attribute (dontRegister) in the DAX.
- all files must be accessible via gridftp to the Pegasus submit hosts that run jobs at the site. Jobs are run at each site in (or below) a "working directory" (**workdir**)
- the **workdir** is created when the site is set up for the user
- a temporary "**job directory**" may be created within the workdir for the duration of the workflow or partition and destroyed when the workflow completes or partition completes.
- A job starts with its CWD set to the jobdir(or the workdir). Hence relative pathnames referenced by a job are relative to this CWD.

Note

Pegasus currently does not create per job temporary directory. It expects the application to create unique files or create unique job directories itself if two or more jobs will produce output with the same file.

Currently a submit host must be able to access any site's sdir, workdir, and jobdir via gridftp. These directories are specified in the site catalog as follows (using XML as the example here):

```
<site handle="chalant" gridlaunch="/home/pegasu/bin/kickstart"
sysinfo="cpuarch">
<lrc url="rls://pegasus.isi.edu"/>
<gridftp url="gsiftp://pegasus.isi.edu"
storage="/home/pegasus" …/>
<workdirectory >/home/pegasus</workdirectory>
<jobmanager ..… />
</site>
```

In XPath syntax we have the following fields of interest:

- sdir = site/gridftp@storage
- gftp = site/gridftp@url
- workdir = site/workdirectory

We also need to define how file names are referenced and processed:

- LFN `a/b/c`

- SFN `<workdir>/a/b/c`
- TFN `<gftp><workdir>/a/b/c`

Pegasus assumes that workdir is accessible, and that sdir is a separate storage location it can use to stage data to/from. Pegasus does not distinguish between internal and external paths to the same data. The external path to workdir is constructed as gftp + workdir.

2.3. Pegasus File Management Conventions

We describe what happens at run time in a Pegasus DAG:

- for setup
- for stage-in
- for compute job filenames
- for stage-out
- for replicas
- for inter-pool transfers
- for cleanup

We also specify related aspects of the compute job (filename arguments - how are they created) and replica registration (what TFN is registered). Given the following site catalog excerpt:

- site 1
  - wdir1
  - sdir
  - gridftp=gsiftp://HOST1
- site 2
  - wdir2
  - sdir2
  - gridftp=gsiftp://HOST2

In addition, Pegasus allows the users to specify a relative path or an absolute path in the properties file that applies to all the sites. Given the following properties excerpt:

- vds.dir.exec *pwdir*
- vds.dir.storage *psdir*

Depending upon whether *pwdir/psdir* is relative or absolute the *wdir/sdir* are either appended or replaced.

Table 1.1.

	pwdir	psdir
XXdir is relative	wdir1 = wdir1 + pwdir	sdir1 = sdir + psdir
XXdir is absolute	wdir1 == pwdir	sdir1 == psdir

All subsequent filename resolutions below assume that the above transformations have taken place.

2.3.1. Setup Job

Pegasus uses a setup job to create the remote *jdir1* from *wdir1* for a partition of a workflow. A partition can be as large as the full workflow, or as small as a single job. By default the full workflow constitutes one partition. The job directory is created by appending a random string to the wdir directory. Thus, the jdir is a function of

- wdir:jdir1 := wdir1 + random

One setup job is created for each execution site, where the portions of a partition have been scheduled. All jobs of the same partition and scheduled for the same site share the same *jdir1*.

2.3.2. Stage In

Pegasus distinguishes between direct (peer to peer) transfers and 3rd-party transfers (3pt). For the replica catalog look-ups, it uses the LFN to find entries. In direct transfer mode, one or more transfer jobs run on the gatekeeper to pull files:

- sTFN := pfn_from_replica_catalog(LFN)
- dTFN := file:// + jdir1 + LFN

In 3rd party transfer mode (set by specifying the property pgs.transfer.thirdparty.sites), the transfers are initiated from the submit host between the remote source servers and destination server:

- sTFN := pfn_from_replica_catalog(LFN)
- dTFN := gsiftp://HOST1 + jdir1 + LFN

2.3.3. Compute Job

All the compute jobs are run in the jdir directory. The *jdir* directory is determined as explained in # Section 2.3.1, “Setup Job” . All references to filenames on the command-line are relative to the jdir1. Since filenames are flattened, the following translation into SFNs applies.

- SFN := jdir1 + LFN

The filenames actually used are stripped of their *jdir1* prefix to generate relative paths to files. The command-line arguments that derive from filenames are resolved relative to the CWD, and contain only the LFN.

2.3.4. Stage Out

The stage-out distinguished between direct (peer to peer) transfers and 3rd party transfers (set by specifying the property pgs.transfer.thirdparty.sites). For direct transfers, files movement is initiated on the remote site between:

- sTFN := file:// + jdir1 + LFN
- dTFN := gsiftp://HOST2 + sdir2 + LFN

In 3rd party transfer mode, file transfer is directed from the submit host between:

- sTFN := gsiftp://HOST1 + jdir1 + LFN
- dTFN := gsiftp://HOST2 + sdir2 + LFN

In the above, the site2 stands in as the output pool where appropriately tagged result files are to be transferred to.

2.3.5. Replica Registration

Files marked for registration are registered with the replica catalog. The Replica Catalog where to register is picked up from the site catalog or the properties file and corresponds to the output site that the user specifies at runtime. The file to register for a given LFN uses the dTFN of the stage-out.

- gsiftp://HOSTxxxx + sdirX + LFN

The replica that is registered in the replica catalog, is the one residing on the output site i.e the replica that was staged to the output site by the stageout job. The replica's that are on the execution sites are not catalogued currently, as they are usually on scratch space that can be purged according to the site's policy or by the cleanup mechanism in Pegasus.

2.3.6. Inter-pool Transfer

Inter-pool transfer jobs are a variant of stage-in jobs, and are created under the following conditions:

- A job X is scheduled at site1.
- Its immediate parent P(X) is scheduled at site2 unequal site1
- Job X requires an input file that is generated by job P(X).

Again, in direct (peer to peer) transfers, the transfer job or jobs are executed on the destination site with the following filename translations:

- sTFN := gsiftp://HOST1 + `jdir1` + LFN
- dTFN := file:// + `jdir2` + LFN

In 3rd party transfer mode, the following transfers are directed from the submit host:

- sTFN := gsiftp://HOST1 + `jdir1` + LFN
- dTFN := gsiftp://HOST2 + `jdir2` + LFN

The jdir directory is determined as explained in Section 2.3.1, "Setup Job"

2.3.7. Clean-up Job

When Pegasus setup jobs created partition-specific remote job directories, a clean-up DAG is generated. The submit files for the clean-up DAG are generated in a cleanup directory in the partition's or workflow's submit directory. The clean-up DAG consists of clean-up jobs for each execution site, where the portions of the partition were scheduled and run. The clean-up DAG is not submitted automatically, and has to be submitted manually by the user.

A new feature in Pegasus also allows cleanup of files from the remote jdir while the workflow is running. This feature is enabled by default and can be turned off by using the option --nocleanup on the command line to Pegasus. This cleanup features adds cleanup jobs at various positions to cleanup files which are no longer needed for further execution of the workflow. This results in effective usage of remote site disk space.

2.3.8. Future Extensions to the Site Catalog

We need the separate storage dir for staging and storing files, and a work directory to run jobs. We need to know, how to access these directories both, from the inside and from the outside. Thus, we need, in addition to what the site catalog provides today,

- a storage dir element for the inside view.
- a workdir gridftp path for outside view of the workdir.

Some sites may not permit outside access to the workdir, or inside access to the storage dir, e.g. LCG2. Such a site requires 2nd-level staging. We will deal with these another time.

```
<directory type="working" url="gsiftp://host1" internal="dir1"
 external="dir2">

<directory type="storage" url="gsiftp://host2" internal="dir3"
```

```
external="dir4">
```

Pegasus constructs the external view of the working directory by adding the site catalog's wdir1 to the gridftp base URI. Pegasus constructs transfer filenames (TFN) according to the XPath syntax from section 1.2.3 as per the following rule

- TFN := pool/gridftp@url + pool/workdirectory+ LFN

The TFNs constructed according to the above rule are listed by transfer job type below:

- Stagein jobs : dTFN
- Interpool jobs : both sTFN and dTFN
- Stageout jobs : sTFN

Bringing in a notion of explicitly defining a gridftp server with the storagedir and workdir will solve this problem: We do not have to use the gsiftp URI from the storagedir to get the outside view to the workdir. The external view of the `wdir1` can now be correctly constructed from the correct knowledge. In XPath syntax from the above excerpt, the resulting filename becomes:

- TFN := directory[@type="working"]@url + directory[@type="working"]@external + LFN

Since we know how to access each file system inside as well as from the outside, it will be a much better and more flexible design.