-- AndreiTsaregorodtsev - 16 Oct 2008
-- StuartPaterson - 03 March 2009

Production Job Finalization procedure

The current production job finalization mechanism is split into four modules: SendBookkeeping; UploadOutputData; UploadLogFile and FailoverRequest. All of the above accept the following "Enable" flag - Boolean True (default) / False. This allows to disable any irreversible actions such as BK records / file uploads etc. whilst printing as much useful information as possible. If no JOBID environment variable exists the Enable parameter is set to False by default (can be useful for testing).

SendBookkeeping

Sends BK records as prepared by the BKReport module. In case of the workflow or step status not being OK this will report to the ProductionDB and exit. Note that the BK replica flags are not set at this point, replica flags are set during the UploadOutputData module.

UploadOutputData

This module establishes the relevant metadata for output files (such as the POOL GUID(s) from local catalogs) and attempts to transfer and register files with failover after resolving the appropriate destination SE. The BK replica flags are set automatically in the case of a successful transfer and are added to a failover request in case of upload failures. If the destination SE is not available files can be transferred to a Tier1-FAILOVER SE (all are attempted).

UploadLogFile

Logs are always uploaded regardless of the workflow status. This module will copy and register files to Grid storage in case of failure and set the appropriate requests to recover at the end. An attempt is made to change the permissions of the files to be readable from the LogSE but this depends on the site specific settings and can fail (printed in the payload logs in this case). Some sites have strange permissions resulting in log files not appearing on the LogSE URL by default so we may need to introduce a server side action there.

FailoverRequest

This module is always last in the chain and creates any pending requests to eventually be submitted by the job wrapper. In the case where the workflow status is OK the FailoverRequest module marks the job as successful. In case of failure pending requests are still submitted.

The following utilities are used in these modules in order to simplify the code:

```
DIRAC.DataManagementSystem.Client.PoolXMLFile - getGUID()
LHCbSystem.Utilities.ProductionData - constructProductionLFNs()
LHCbSystem.Utilities.ResolveSE -  getDestinationSEList()
```

The 2 distinct results of the finalization are explained below:

- application part finished successfully;
- application part failed

Case 1. Application successful

1. Bookkeeping records (not replica flags) are sent to the Bookkeeping service prior to uploading the data files. If any of the bookkeeping records sending fails a corresponding failover request is created.

2. Output data upload. For each output file the destination Storage Elements are resolved according to the job workflow parameters. The upload to the specified destination is attempted with registration in all the DIRAC catalogs (LFC, BookkeepingDB(replica flag), ProductionDB). If upload fails, the file is uploaded to one of the FAILOVER storages and the corresponding failover request is created.

If several destinations are specified for a given file, they are attempted in turn until the first successful upload. In this case replication requests are created to copy the file to other specified destinations. If all the specified destinations fail, the file is uploaded to one of the FAILOVER storages and the corresponding failover replication requests are created.

The upload to the FAILOVER storage is only registered in the LFC catalog.

If for at least one output data file neither upload (destination or failover) is successful:

- the job is declared Failed;
- all previously defined requests for other output files are dropped (except for log files failover request if any);
- data removal requests are created for already uploaded files;
- the input data files are set to "Unused" in the ProductionDB;
- the already sent bookkeeping records have no replica flags.

3. Log files are uploaded to the Log Storage Element. If the upload fails, the log files are tarred and put into the Failover system. This is always performed regardless of the workflow or step status in order to ensure logs availability.

4. The combined request is written into a file to be picked up by the Job Wrapper

Case 2. Application failed

1. Log files are uploaded to the Log Storage Element. If the upload fails, the log files are tarred and put into the Failover system. This is always performed regardless of the workflow or step status in order to ensure log availability.
2. Input files status is updated in the Production DB Service as "Unused" except for files marked as "ApplicationCrash" by the application log analysis module. If the update fails, the corresponding failover request is created.
3. The combined request is written into a file to be picked up by the Job Wrapper.

This topic: LHCb > ProductionJobFinalization
Topic revision: r5 - 2009-03-04 - StuartPaterson