

ProductionAgent BulkSubmission

Specification

The bulk size for multiple job submission is specified through a configuration parameter defined in file *ProdAgentConfig.xml* with default value 1, which can be changed in running time by sending the message *setBulkSize*.

A single tar file is defined for all jobs submitted in a single bulk. The tar file contains the specification for the first job in the sequence. It includes a script that is executed in the worker node, which updates the values of the configuration parameters (.cfg file) to make them adequate for the particular bulk instance.

Each job has a corresponding entry into the *JobStateInfo* tables, it does not matter if it is a job submitted independently or a single instance of a job submitted in a bulk.

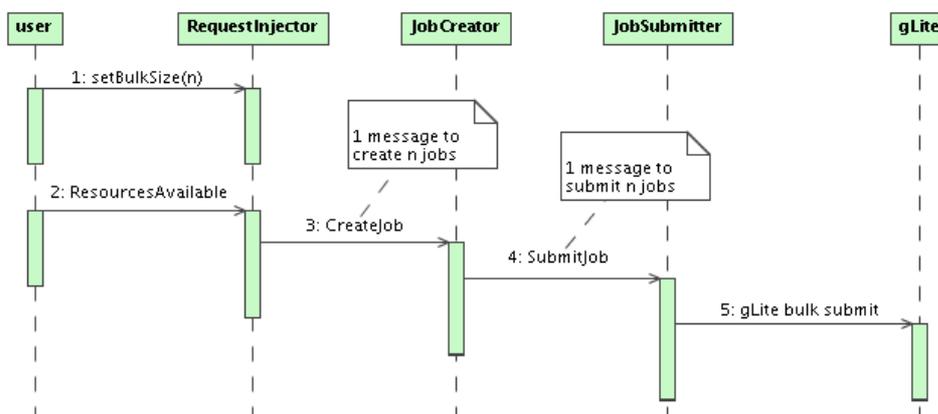
Cache cleanup is started only when all jobs in the bulk are in a combination of finished successfully and/or general failure status. In other words, the shared tar file is removed when no job in the bulk can be resubmitted, because they have finished successfully, or they have reached the maximum number of failures.

As an initial simplification, in case of failure, a job is resubmitted individually, even if it was originally submitted inside a bulk. It is not difficult to resubmit in bulk, but let us start simple. From the implementation point of view, the job specification for single job resubmission has to be parametric, since the same tar file is reused.

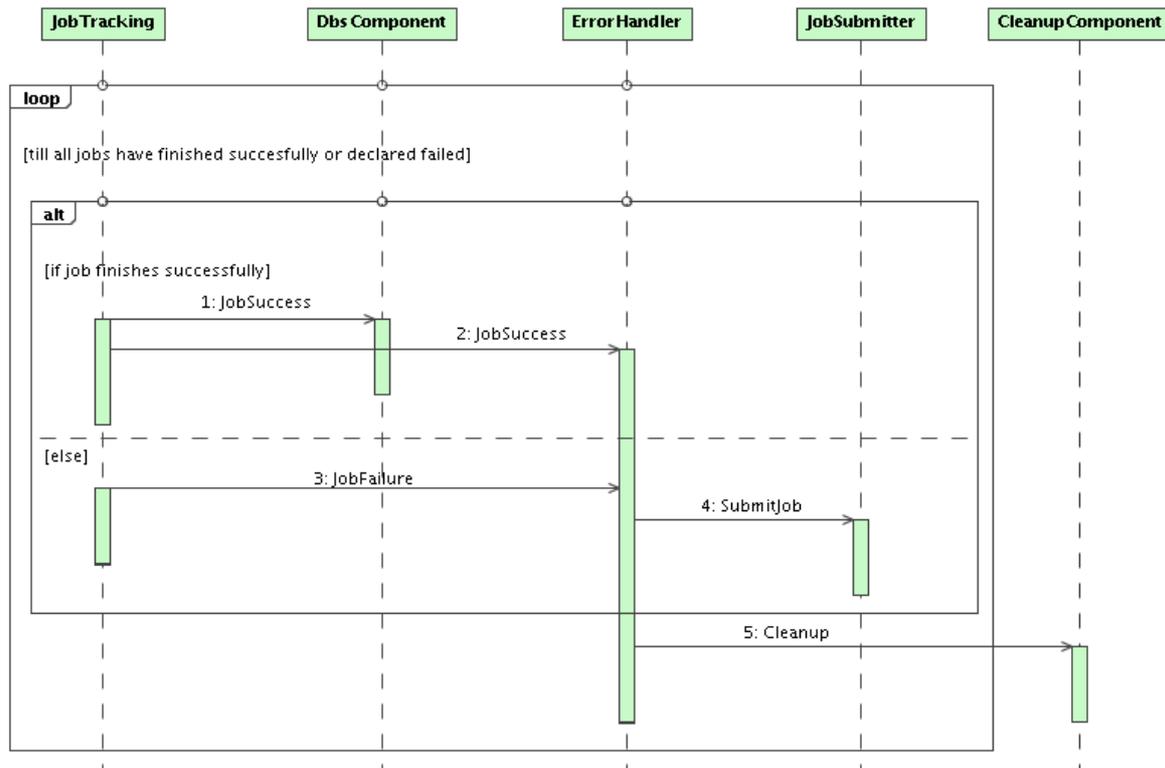
With gLite, we have the concept of main bulk job, which is not really a job, but it is the name of the bulk of jobs. I particularly do not like this concept *in the context of the Production Agent*, since these jobs do not share most properties with all other jobs (for example, there is no framework job report, etc). In order to have a transparent integration with the Production Agent concept of jobs, I propose that the main bulk job has to be a hidden concept inside gLite plugins and tracking component.

Sequence diagrams

Job Creation



Cleanup



Implementation

Requirements

- Add message `setBulkSize(<size>)` used to set the parameter `bulkSize` in the job specification. Its default value is 1.
- The `ResourcesAvailable` message now increments the internal iterator counter by `bulkSize`.

Product DB

- Add the boolean field `BulkJob` to the table `js_JobSpec` to indicate if the job was submitted in a bulk of jobs or as a standalone job.
- Create the table `js_BulkJobs` with the fields `JobSpecID`, `ParentJob` and `Index`. This table keeps one entry for each job submitted in bulk, specifying the id of the job, a reference to the first job in the bulk and the index number in the bulk sequence (0, 1, 2, etc.).

JobCreator

- Define the method `supportBulkCreation()`, which returns `False` by default in the `CreatorInterface` metaclass.
- Override the method `supportBulkCreation()` in the class `LCGCreator` to return `True`.
- Add check in `JobCreator` that when `bulkSize` is not 1, the current plugin has to support bulk creation.
- Add loop to register all individual jobs in `JobStateInfo`, with the same cache area.
- Add in Glite plugin the inclusion of file `updateConfig.py`, which must be executed after `RuntimePSet Prep.py` in order to replace the values of `JobName` and `RunNumber` (at least) by the value of `_PARAM_` + the value specified in the configuration file. As an example, if the configuration file specifies 55 as `RunNumber`, the modified configuration file will have the value 55 in the first job, 56 in the second, 57 in the third, etc.

JobSubmi t t er

- The gLite plugin must generate the jdl specification by specifying *StartParameter* as 0 and *Parameters* as the *bulkSize*. In case of resubmission, the value of *StartParameter* is the job index and the value of *Parameters* is 1.

Tr i gger

- Modify the method *TriggerAPIMySQL.setFlag()* such that it returns *True* only when all flags associated to members of the bulk are set to '*finished*'. In other words, the associated action will be triggered only when all jobs in the bulk would have finished their associated flags (by success, general failures, etc.). The behavior is not modified for non bulk jobs.
- Modify the method *TriggerAPIMySQL.resetFlag()* to reset the flag to 'start' value, following the same restrictions as currently implemented in boolean terms, by checking also bulk conditions.
- Modify the cleanup trigger handler to remove also the entries in the new table *js_BulkJobs* associated to the bulk of jobs that has to be cleaned up.

JobTracki ng

Incomplete, left to experts.

- Take care of getting the files produced by the jobs in a proper directory.

-- CarlosKavka - 03 Oct 2006

This topic: [Main > ProductionAgent BulkSubmission](#)

Topic revision: [r1 - 2006-10-03 - CarlosKavka](#)



Copyright © 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback