LOFAR batch scheduler interface

17-10-2013 (Wouter Klijn, Bob Droge, Alwin de Jong)   
version 0.2

The goal of the LOFAR batch scheduler interface is to describe and formalize the communication between two to be created software systems, the DPU batch scheduling framework and a LOFAR side pipeline management and control module.

The communication will be based on a subset of the LOFAR parset configuration language that is compatible with xml. This will prepare for future switch to a fully xml based control mechanism.

The interface should unambiguously describe individual pipelines settings, the processing locations, the input data, the dependencies between pipelines, and dpu settings. It will further describe the feedback as produced by the DPU detailing the success or failure of the runs and other information of interesting.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | | | Value (examples) | Comments |
| pipeline\_block. | pipeline\_block\_id | | GUID | Globally Unique ID for the pipelineblock |
| pipeline\_type. | pipeline\_id | Observation123456 | Obs-id |
| config | Parset/xml tree | LOFAR produced parset |
| inputs | [input\_calibrated] | Key into the parset/xml config tree |
| outputs | [output\_instrument] | Idem. |
| predecessors | [observation654321, Guid] | Wait for other pipeline to be finished |
| dpu\_pipeline \_config | Parset/xml tree | Finegrained control of dpu functionality, not needed in pilot |
| comments | “Create by John Doe” | not needed in pilot |
| pipeline\_type2. | … | … | Next pipeline |
| dpu\_block\_config | | Parset/xml tree | Not needed for pilot project |
| comments | | “test 3” | not needed in pilot |

Figure 1: hierarchical structure, keys names, example values and comment detailing the input parset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | | | Value (example | Comments |
| pipeline\_block. | pipeline\_block\_id | | GUID | Globally Unique ID for the pipelineblock |
| receive\_time | |  | Time input parset was received |
| start\_time | |  | Time first item was send |
| end\_time | |  | Time last item was finished |
| return\_status | | ok/fail/aborted | Abort is stopped due to eg. time constraints or user |
| pipeline\_type. | pipeline\_id | Observation123456 | Obs-id |
| return\_status | ok/fail/aborted |  |
| start\_time |  | Start first item of list |
| stop\_time |  | Finish of last item in list |
| job\_return\_status | [ok, ok, fail, fail, abort] | For each work item the return state. Not needed for pilot |

Figure 2: hierarchical structure, keys names, example values and comment detailing the output parset