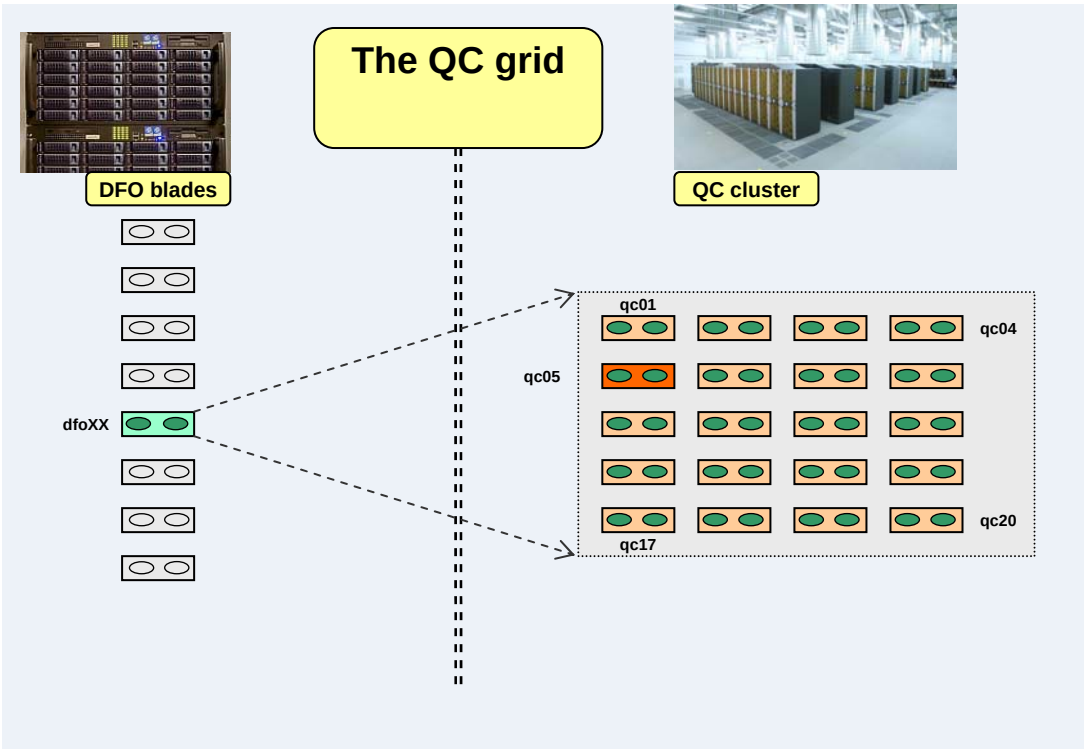


The QC grid

Version 1.0 Reinhard Hanuschik 2010-06-01

Note: see also the tool documentation under http://www.eso.org/~qc/dfos/XportJob.html.

The QC cluster consist of 20 identical dual-core blades. With 2 of them reserved for interactive QC work (`hawki` and `vircam`) and one reserved for managing condor („`condor_master`“), there remain 17x2 = 34 cores for condor execution. This compares to 2 cores available on a dfo blade for condor execution plus all foreground (e.g. certification) or background (e.g. *trendPlotter*, *qc1Parser*) jobs for a given instrument.

The QC cluster is currently saturated by VIRCAM processing and QC reports for roughly 24 hours when new data disks arrive on Tuesdays. Outside that time window, the cluster is largely „idle“ in the sense that all other dfos workflow steps cannot be run in parallel[1]. Also, most of them scale in terms of performance by number of files, not by file size. In other words, the QC cluster has plenty of CPU cycles to offer outside the VIRCAM processing window.[2] The current activity on the QC cluster, plus the submitted queue still to execute, is monitored on the „cluster monitor“ (http://safweb1/~qc/CLUSTER/monitor/clMonitor.html).

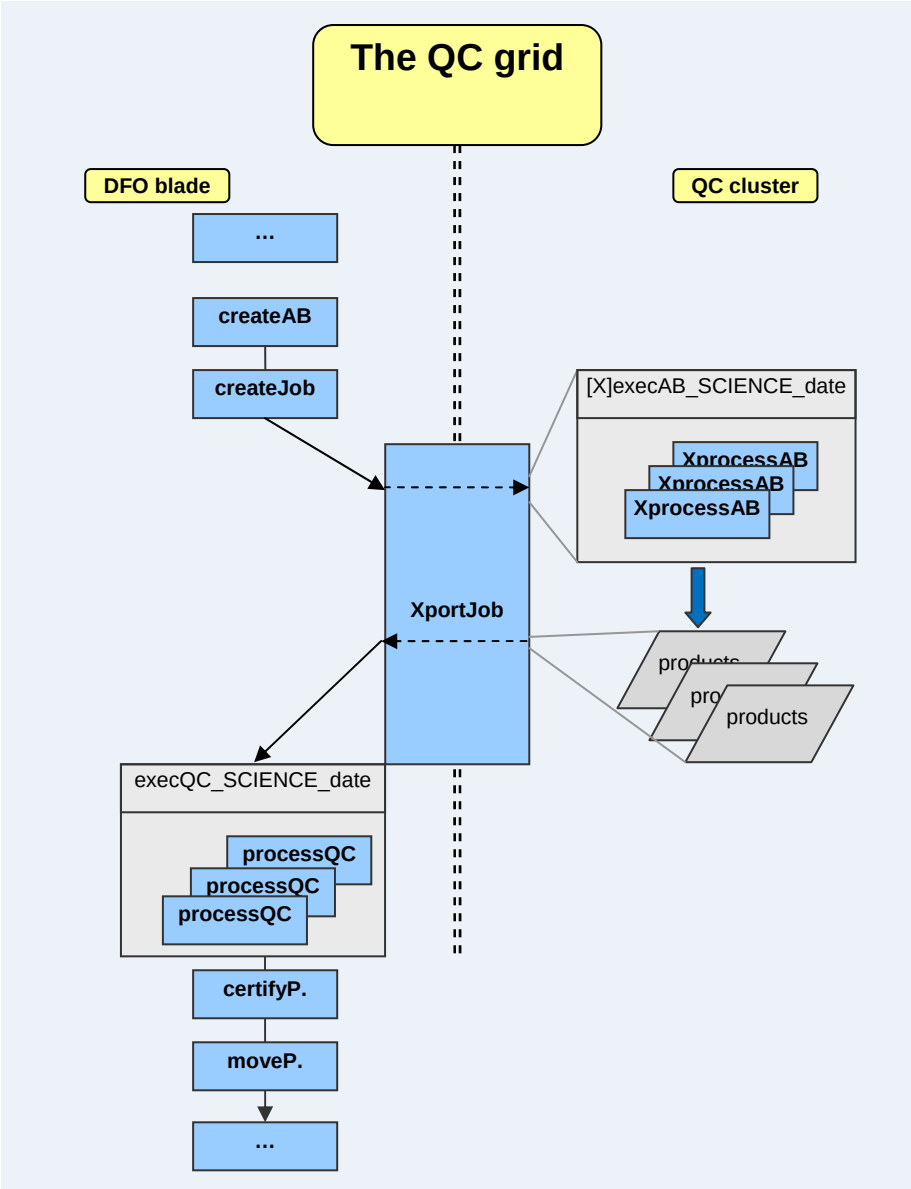
It is therefore reasonable to use the QC cluster as compute platform whenever it has empty CPU cycles to offer. In principle this can offer processing by a factor 17 (= 34/2) faster, neglecting overheads for data transfer. This scenario is particularly attractive for large, self-contained data sets. Self-contained means having dependencies only on archived calibration data, or on non-archived (virtual) calibrations if created by the same cascade. Prime candidate data sets are:

- Science data from burst mode or other large volume nights („visitor mode nights“)
- Science data for reprocessing projects
- Calibration data that need to be reprocessed from historical data sets

Not so well suited are:

- Smaller science data sets from normal nights (since there are overheads)
- Normal calibration data sets (since there are dependencies).

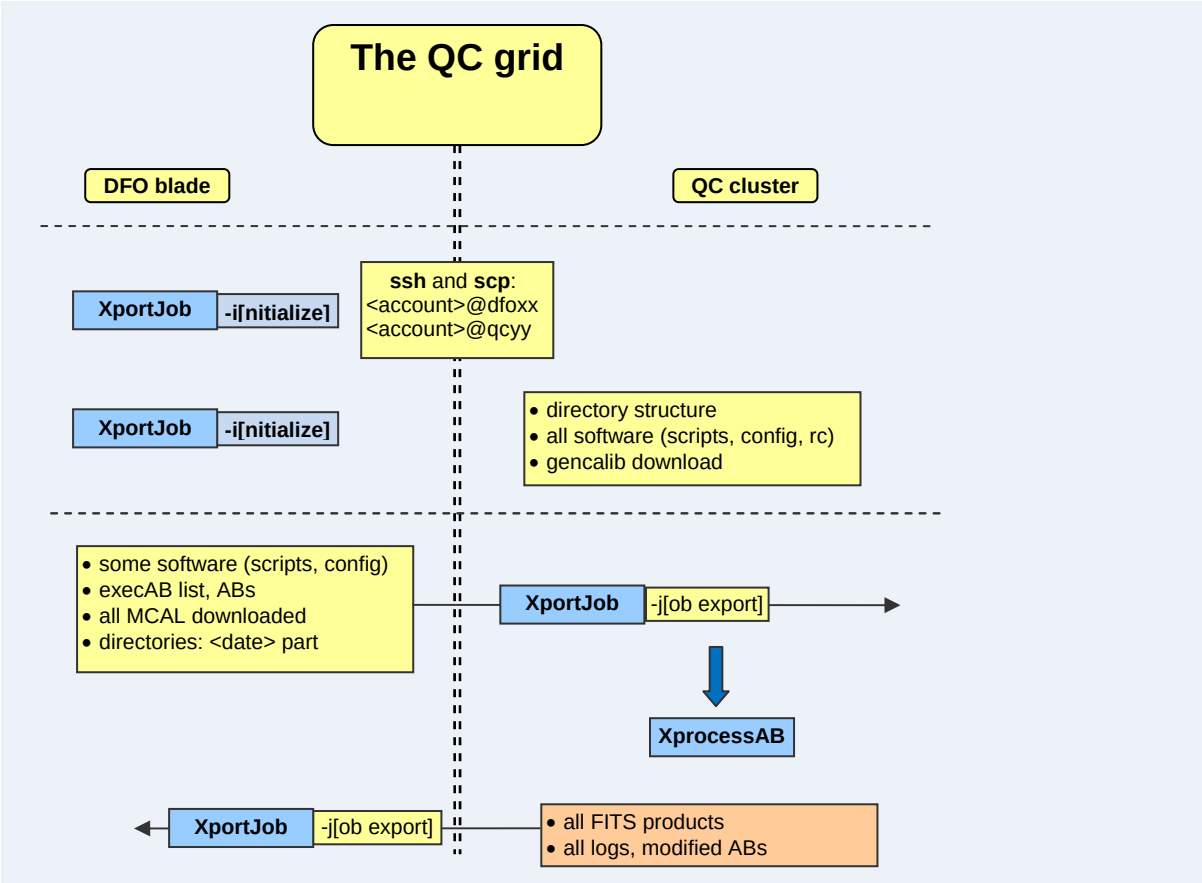
[1] The HAWKI processing is neglected here.
[2] This will in principle not change with the onset of OCAM processing.



Workflow

The standard dfos workflow needs to be modified in one place. In the jobs file, there is an entry starting with “vultur_exec_cascade“ which launches the condor execution of the processing job on the local dfo blade. Clearly, that script would also launch the job on a QC blade. But there are other workflow steps needed, mostly related to data download and data transfer (see below). Hence, it is reasonable to replace “vultur_exec_cascade” by a new workflow tool, **XportJob**.

This tool creates on the fly a few other modules for execution on the QC cluster, and also creates a wrapper around processAB. That wrapper (XprocessAB) and the helper modules all start their names with capital **X** (for **export**) to mark their membership to the same family of tools for the QC grid.



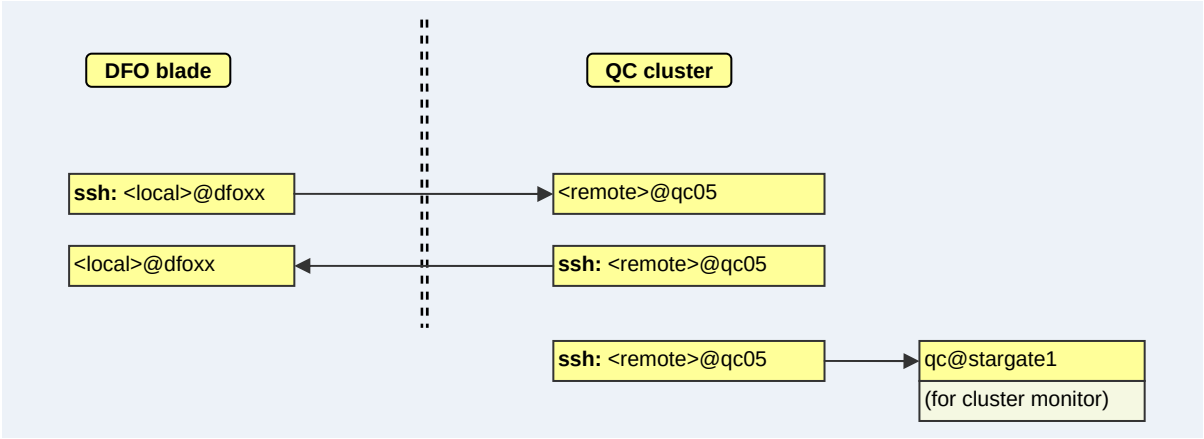
Tool XportJob

The tool XportJob is the main tool to export and execute compute jobs on the QC grid.

It can be used in two ways:

- To initialize, call option –i: this will test and setup ssh access to the QC cluster, create the required directory structure, export the required software (tools and configuration), and fill the general mcalibs.
- To operate, call option –j <job_file>. The job_file is the one created by createJob. It contains a set of ABs. The cascade needs to be self-contained: either no dependencies, or only dependencies from mcalibs, or dependencies within the same cascade. For historical reprocessing, you can also copy the ABs into $DFO_AB_DIR, then call options –d <date> -m <mode> and thereby create the job file.

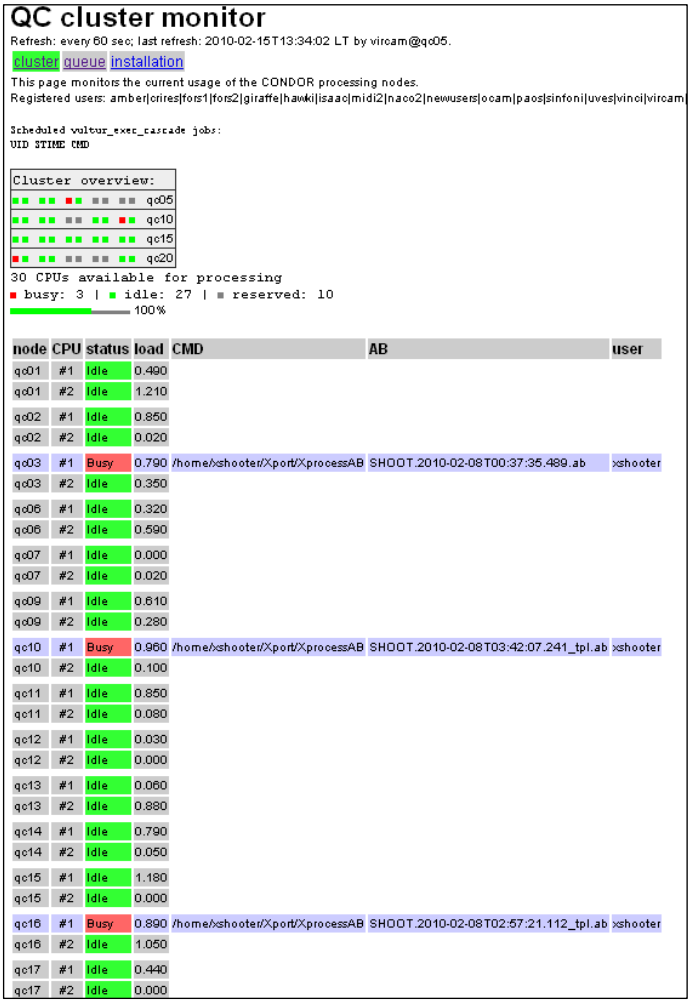
The tool will deliver all products (fits, ABs, logs) as if the cascade had been executed locally. They can be used for the QC reports and then for certification.



ssh access

Overview of required ssh access from local account@dfo_host to remote account@qc05 and vice versa, and from remote account@qc05 to qc@stargate1.

Cluster monitor



QC cluster monitor

Refresh: every 60 sec; last refresh: 2010-02-15T13:34:02 LT by vircam@qc05.

cluster queue installation

This page monitors the current usage of the CONDOR processing nodes.
Registered users: amber|crires|fors1|fors2|giraffe|hawki|isaac|midi2|naco2|newusers|ocam|paos|sinfoni|uves|vinci|vircam|

Scheduled vultur_exec_cascade jobs:
UID STIME CMD

Cluster overview:
qc05
qc10
qc15
qc20

30 CPUs available for processing
busy: 3 | idle: 27 | reserved: 10
100%

node	CPU	status	load	CMD	AB	user
qc01	#1	Idle	0.490			
qc01	#2	Idle	1.210			
qc02	#1	Idle	0.850			
qc02	#2	Idle	0.020			
qc03	#1	Busy	0.790	/home/xshooter/Xport/XprocessAB	SHOOT.2010-02-08T00:37:35.489.ab	xshooter
qc03	#2	Idle	0.350			
qc06	#1	Idle	0.320			
qc06	#2	Idle	0.590			
qc07	#1	Idle	0.000			
qc07	#2	Idle	0.020			
qc09	#1	Idle	0.610			
qc09	#2	Idle	0.280			
qc10	#1	Busy	0.960	/home/xshooter/Xport/XprocessAB	SHOOT.2010-02-08T03:42:07.241_tpl.ab	xshooter
qc10	#2	Idle	0.100			
qc11	#1	Idle	0.850			
qc11	#2	Idle	0.080			
qc12	#1	Idle	0.030			
qc12	#2	Idle	0.000			
qc13	#1	Idle	0.060			
qc13	#2	Idle	0.880			
qc14	#1	Idle	0.790			
qc14	#2	Idle	0.050			
qc15	#1	Idle	1.180			
qc15	#2	Idle	0.000			
qc16	#1	Busy	0.890	/home/xshooter/Xport/XprocessAB	SHOOT.2010-02-08T02:57:21.112_tpl.ab	xshooter
qc16	#2	Idle	1.050			
qc17	#1	Idle	0.440			
qc17	#2	Idle	0.000			

The tool *clMonitor* is a visualization of the condor command condor_q. It displays: the current status of the QC cluster (top: green for idle cores, red for busy cores); the current job list of condor (the processes running on the busy cores); and (under ‘queue’) the list of remaining tasks. The tool is called every 60 sec from the vircam account, plus on demand by XportJob.

You can safely launch a new job if the cluster monitor indicates an empty queue.

What is required

To execute jobs on the QC grid, you need to have your operational account registered as a QC grid account. To register your account to the QC grid, you need:

- an account on the QC cluster
- the tools XportJob and clMonitor
- to initialize that account

By begin of 2010, all then operational dfo blade accounts have been installed as an account on the QC cluster. These accounts (e.g. xshooter@qc05) come per default with the same password as your dfo account. Type 'ssh <my_account>@qc05' to verify. They should also have .pecs and .ssh installed.

The tools `XportJob` and `clMonitor` are not yet dfos-distributed. Currently you need to install them by hand by downloading the tar balls from http://www.eso.org/~qc/dfos/.

To initialize an account for the QC grid you need to do the following:

- install `XportJob` and `clMonitor` on <my_account>@dfoXX
- update config.XportJob to contain your account name
- call 'XportJob –i' to start the initialization.

The initialization has the main steps:

- set up and test the ssh connections (dfoXX to qc05 and vice versa; qc05 to qc@stargate1)
- create basic .dfosrc
- create workspace on qc05 (minimum set of dfos directories)
- export required software (dfos tools and configuration)
- export pipeline configuration (the pipelines are installed on the QC cluster in the same way as on the dfo blades; you may want to use the same or a different version)
- populate the general mcalibs to qc05

Once initialized, your account is ready for processing on the QC grid.

Processing on the grid has the following steps:

- provide all necessary ABs (either from a previous createAB call; or by copy from $DFO_LOG_DIR if you want to reprocess) in $DFO_AB_DIR
- provide the job execution file for all these ABs in CONDOR syntax (either exists from a previous createJob call, or can be created with 'XportJob –d <date> -m <mode>'); the job execution file has the name 'execAB_<mode>_<date>'. In most cases it will be of mode SCIENCE
- call 'XportJob –j execAB_<mode>_<date>'

Remember that the cascade in a job file must be self-contained. Typically there are no dependencies on virtual calibrations at all. At least there are no virtual calibs outside the cascade.

The tool will first refresh the software (there could have been updates since the initialization). Then, all required raw files are either transferred from the dfo blade to the QC cluster (if existing), or downloaded from NGAS into the QC cluster (if not). Then the same happens to mcalibs. These bulk downloads are required before processing since otherwise many simultaneous NGAS requests would be triggered by the `processAB` calls on the QC cluster.

Finally the job file is submitted to condor, and processing on the QC cluster takes place, with up to 34 nodes.

After an AB has finished, all its products, plus the AB and the processing log, are transmitted back to the dfo blade. After the cascade has finished, all raw and mcalib data (except for the general ones) are

deleted or moved back to the dfo blade. By default, all data (fits files, logs, condor logs) are deleted on the QC cluster (“*leave nothing, not even footprints*”).