

Virgo Data Quality Report

These pages describe the data quality report (DQR) system used to vet GW candidates.

[Virgo Data Quality Report](#)

[Overview](#)

[Getting started](#)

[Technical information](#)

[Alert reception](#)

[GraceDB upload](#)

[Tests](#)

[O2 replay](#)

[Review](#)

[Documentation](#)

[Description of the checks](#)

[Instructions for shifters / RRT members](#)

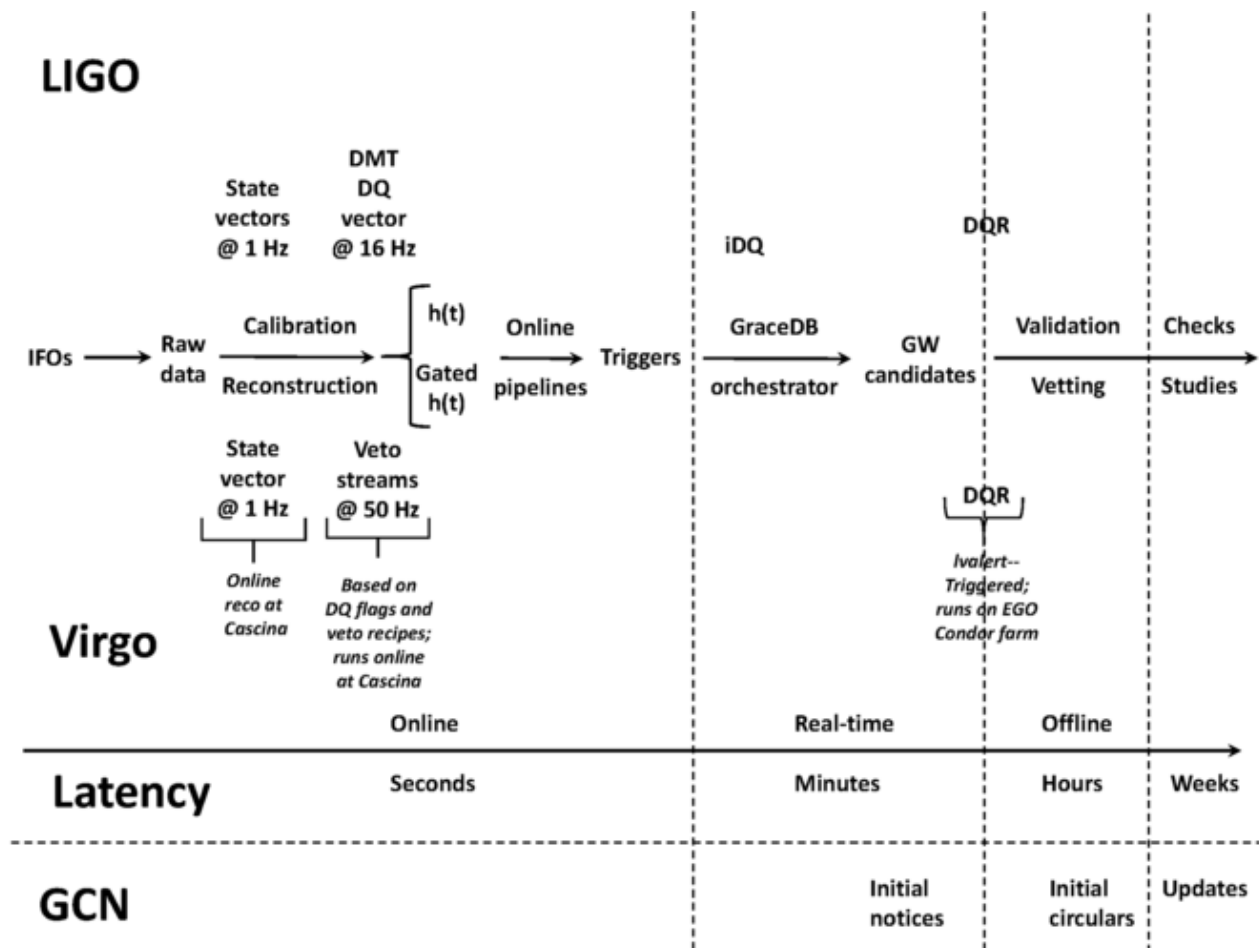
[More expert information](#)

[How to add a check to the VirgoDQR package](#)

[FAQ](#)

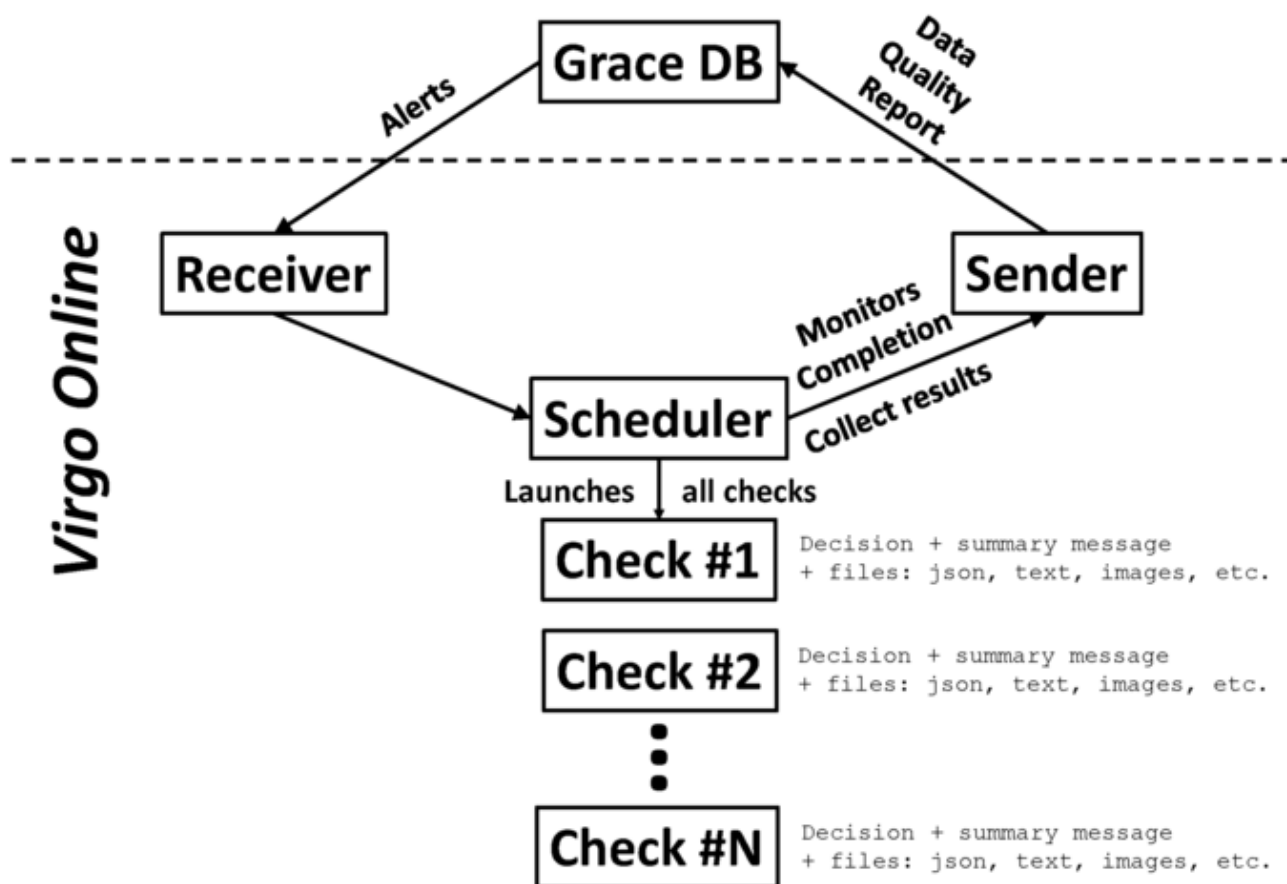
Overview

During O3, more GW events are expected due to the detector sensitivity improvements. Moreover, all alerts will be public ('OPA' era) and most of them should be sent automatically. Therefore, it will be important to assess quickly the quality of each GW candidate found by online pipeline -- if problems are found, a retraction may be issued to cancel the alert. Therefore, the generation of data quality information for the events has been automated: data quality reports (DQRs) will be produced automatically at the three sites (Hanford, Livingston and Cascina) and their results will be merged in GraceDB to produce joint DQRs.



The online dataflow, including LIGO and Virgo DetChar processes

The Virgo DQR framework is implemented at EGO on the [Condor](#) farm. Whenever a GW alert is received from GraceDB, a new Virgo DQR is triggered. Various scripts are run in sequence, leading to the generation of a Condor DAG that includes all the data quality checks to be run around the time of the event. This DAG is automatically executed: the results of each check are made available in a local webpage based on LIGO's data-quality-report framework and they are uploaded to GraceDB as well, in order to fill a joint LIGO-Virgo DQR.



Zoom on the Virgo DQR

The module `scripts/monitor_condor_dag.py` monitors the evolution of the Condor DAG: its output is available from the local webpage described above. In case of problems, the checks can also be run by hand: various scripts (including some created during the DAG generation phase), are available for that. See below for details.

Getting started

Technical information

The DQR package is named **VirgoDQR** and can be downloaded from the [Virgo SVN repository](#).

```
svn co https://svn.ego-gw.it/svn/advsw/VirgoDQR <tag>
```

with `<tag>` being either an existing SVN tag (check [VPM](#) to see which one is used to run online) or `trunk` to get the development version of the package.

To setup the package,

```
cd <where your package is located>
cd cmt/
cmt config
source setup.sh
make
```

The bash script

```
scripts/generateDAG.sh <target subdirectory> <Event GPS time> [<Event duration>]
```

generates the condor DAG, including all the DQ checks to be run on the EGO Condor farm. Therefore, all checks are described in that script. The target sub directory is relative to the base DQR output directory, currently `/data/procdata/web/dqr/`

Then the dag can be sent either manually or using another bash script from the submit1 Condor machine.

```
Scripts/condor.sh <target subdirectory> <full path of the target subdirectory>
```

The target subdirectory is also the ID of the data quality report for that event. Using the above script allows one to start the monitoring of the DAG running on Condor.

The DQR target subdirectory should be available through the URL [DQR web report](#). Once found (there are subdirectories, including one for O3), click on it and open the `dqr.html` webpage.

Alert reception

Alerts are received from the [!GraceDB database](#) through the [LIGO lvalert system](#). `lvalert_admin` is used to subscribe to channels; `lvalert_listen` is used to listen to them and get alerts. On the different channels, we are receiving new events, plus any update about past events (maps, etc.). For the DQR, we only react on `type of event = 'new'`, generate the dag and run it on the Condor farm through the `submit1` machine. As stated above, the dag running is monitored.

The production version of the DQR is run in VPM: https://vpm.virgo.infn.it:40000/main.html?subsystem=DetChar&process=lvalert_virgo.

GraceDB upload

Virgo DQR results are stored locally and available through a webpage that allows one to navigate among the different checks. In parallel, they are uploaded back to GraceDB in order to make joint LIGO-Virgo DQRs. Whether results are uploaded back to GraceDB or not is driven by the `-u/--upload` option of the `generateDAG.sh` script. The Virgo DetChar robot certificate is used to get access to the database.

Tests

O2 replay

See [this page](#) for details.

Review

See [here](#).

Documentation

Description of the checks

See [here](#).

Instructions for shifters / RRT members

See [here](#).

More expert information

See [here](#).

How to add a check to the VirgoDQR package

As a pre-requisite you should have a working script taking among its input parameters the GraceDB info (currently only the event GPS time). If the script is included in the VirgoDQR package, you can write it so that it produces outputs directly compatible with the [LIGO data quality report](#) framework which we are using for the Virgo DQR reports as well. Otherwise, you will need a second script, running after the first one (Condor will do that for you with its parent/child ranking) and parsing the outputs of your main script to produce information in the right format.

The simplest then is to contact Nicolas Arnaud, Nicolas Leroy or Florent Robinet who can include your check into the Virgo DQR. Otherwise you can try to do that on your own using [these instructions](#) -- feedback welcome!

FAQ

See [here](#).

-- Main.Robinet - 17 May 2018

Center text

Attachments 2

This topic: DetChar > WebHome > DetCharDQR

Topic revision: