

Omicron is now a separate package relying on GWOLLUM libraries

A stable version of Omicron is tagged as v0r1. This is the official tag for ER3.

The packages GWOLLUM/Omicron/UPV are ready to be installed under /virgoApp

Last December, GWOLLUM went under major upgrades:

- The trigger definition/convention changed. Previous trigger objects are now obsolete
- The clustering has been greatly improved (~ 100 times faster)

Omicron is currently running online over ER3 h(t) streams (LIGO+Virgo)

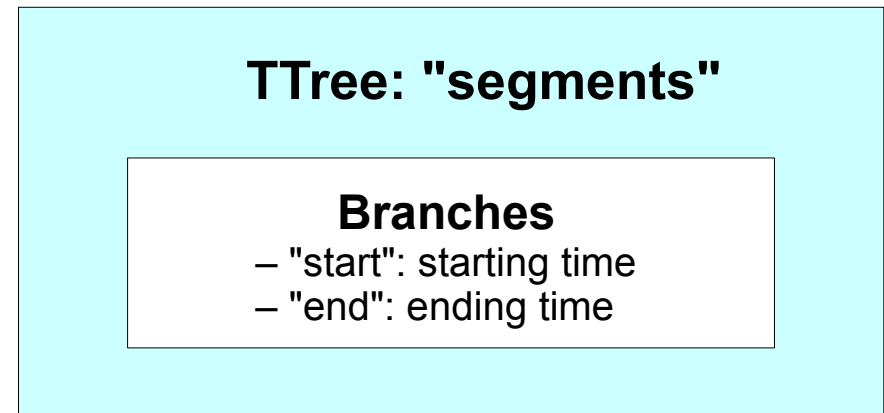
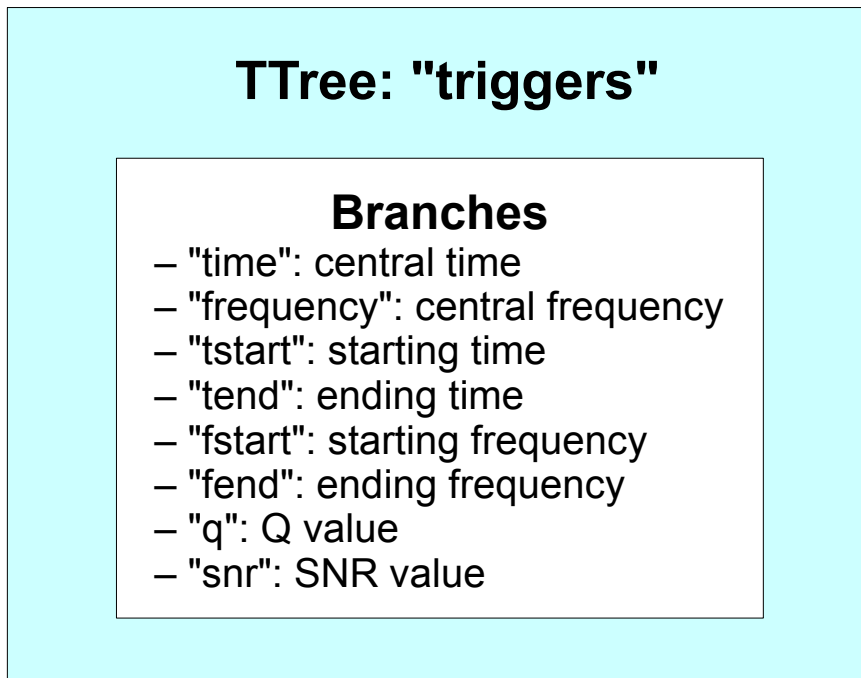
Monitoring web pages: <https://wwwcascina.virgo.infn.it/DataAnalysis/Omicron/index.html>

VSR2 data are currently reprocessed

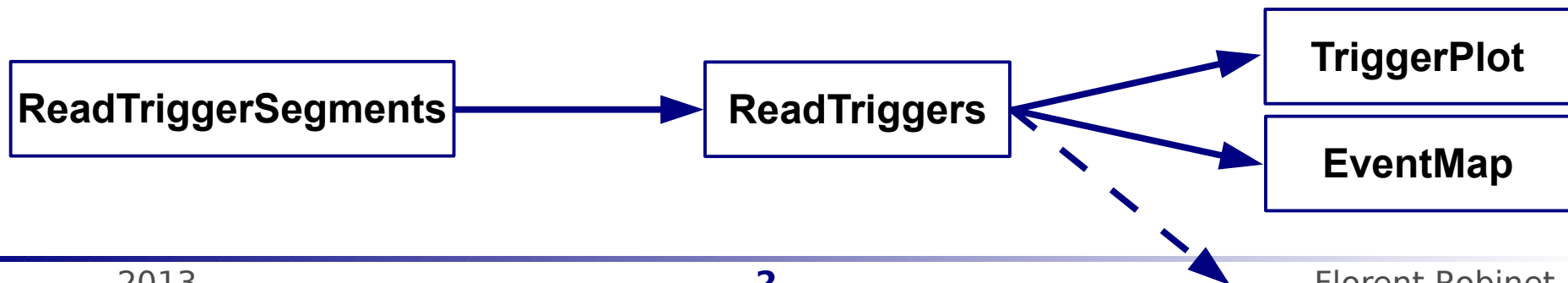
Omicron: Triggers

Omicron produces triggers following the "GWOLLUM convention":

A trigger object is a ROOT file containing 2 **TTree** objects:



Any trigger object following this convention can be read by GWOLLUM functions.
Classes inheritance scheme:



Omicron: Triggers and Clusters

Omicron triggers are un-clustered. This means that we save **EVERY** tile which fires above the SNR threshold.

This is crucial for detchar since it provides a lot of information to study glitch properties

The price to pay is two fold:

1/ this is a lot of triggers!

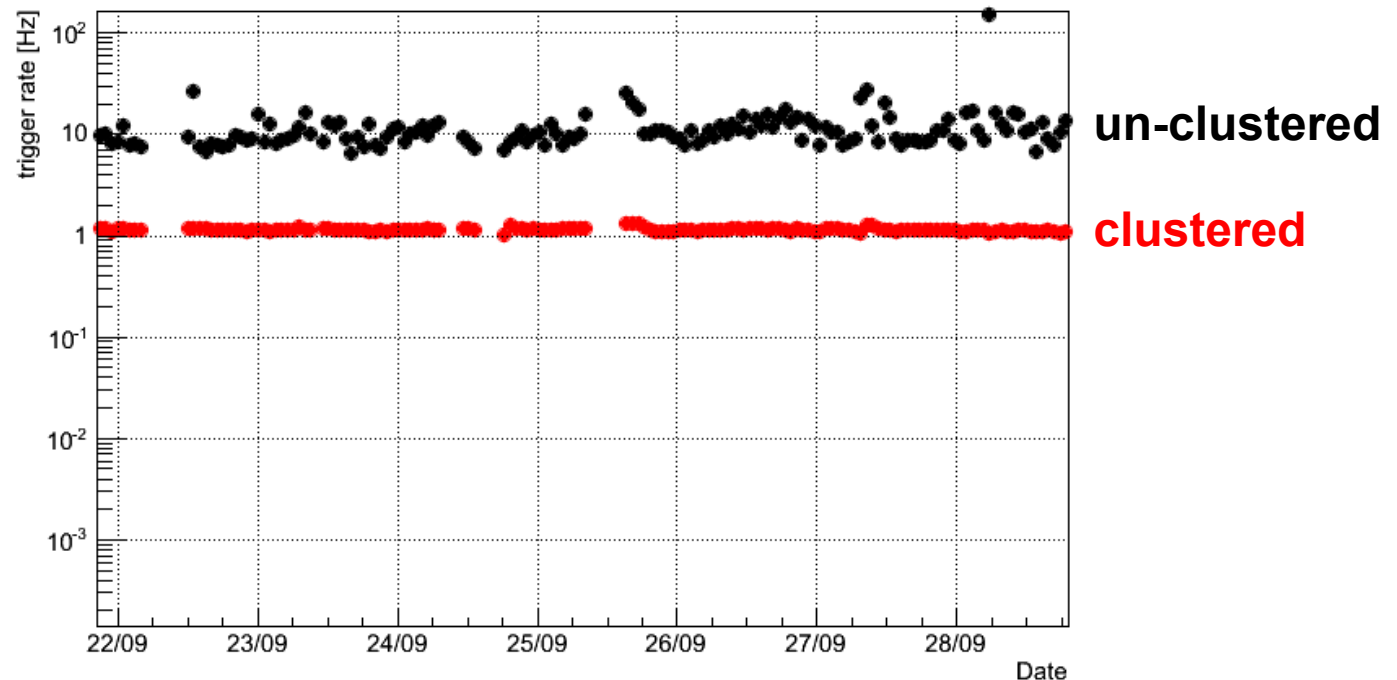
2/ clustering must be performed every time the triggers are used

But this price is not too expensive:

– it has been shown that we can handle this volume of data

– the clustering algorithm has been greatly improved. This step is no more a limitation

Trigger Rate (starts at Sep 21 2009 20:28 UTC)



Typical example:
VSR2 h(t) with SNR>5.5

Omicron triggers can be used in many applications.

It is strongly suggested to use GWOLLUM libraries to access triggers. This is much more efficient!

For reluctant people, it is possible to print the triggers in ASCII format:

```
source /virgoDev/Omicron/v0r1/cmt/setup.(c)sh
```

```
GetOmicronTriggers -c FAKE_h_16384Hz_4R 1043130200 1043130300
```

Omicron: Applications (II)

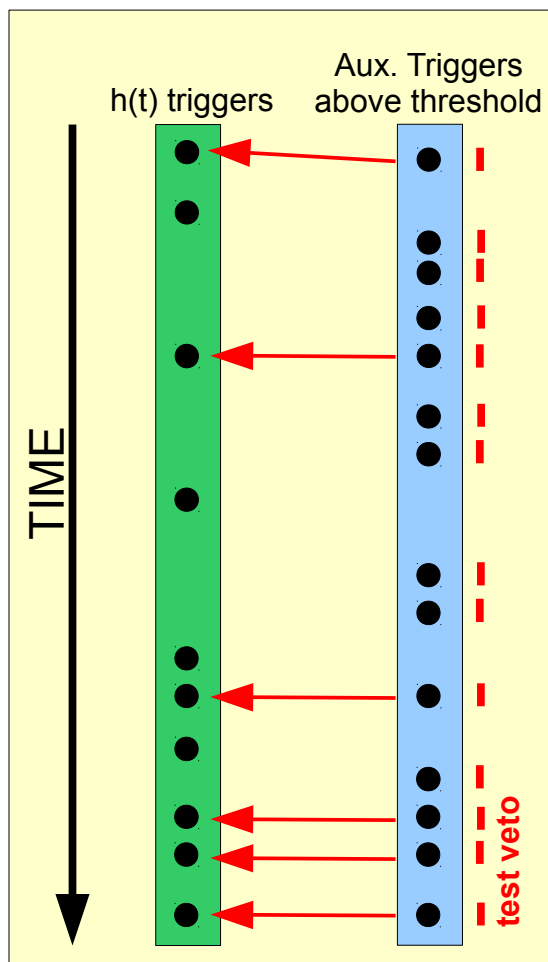
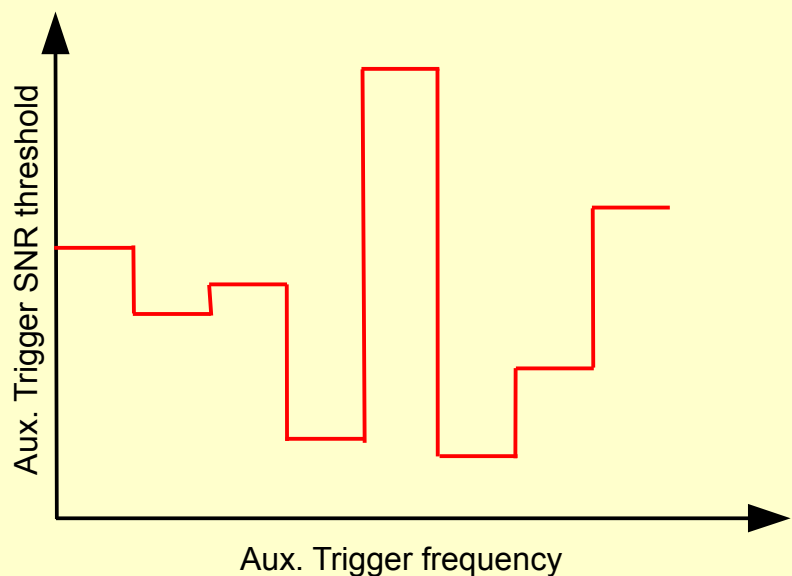
Omicron triggers can be used to produce vetoes.

A GWOLLUM library was created for that purpose: **GlitchVeto**

The principle is simple:

It is based on the time coincidence between an auxiliary channel and the $h(t)$ channel

Here is the most generic way to define a veto on single channel triggers:



Veto recipe:

- you vary your threshold
- you count the coincidences
- you define a FOM to test your veto
- if your veto is not good enough, you raise your threshold



UPV

Currently tested on LIGO ER3 data
N. Christensen

Omicron: Online implementation

Omicron triggers can be produced online

Package: **OnlineOmicron** (maintained by N. Leroy)

Tests are performed on ER data

(data availability) + (data buffering) + (Omicron processing) = Latency

N_{\max} channels

<2s

16s

<12s

< 30s



Will determine how many cores we will need (current estimate = ~30)

Open discussion:

We need to design an online architecture for applications using Omicron triggers