Low Latency data and analysis

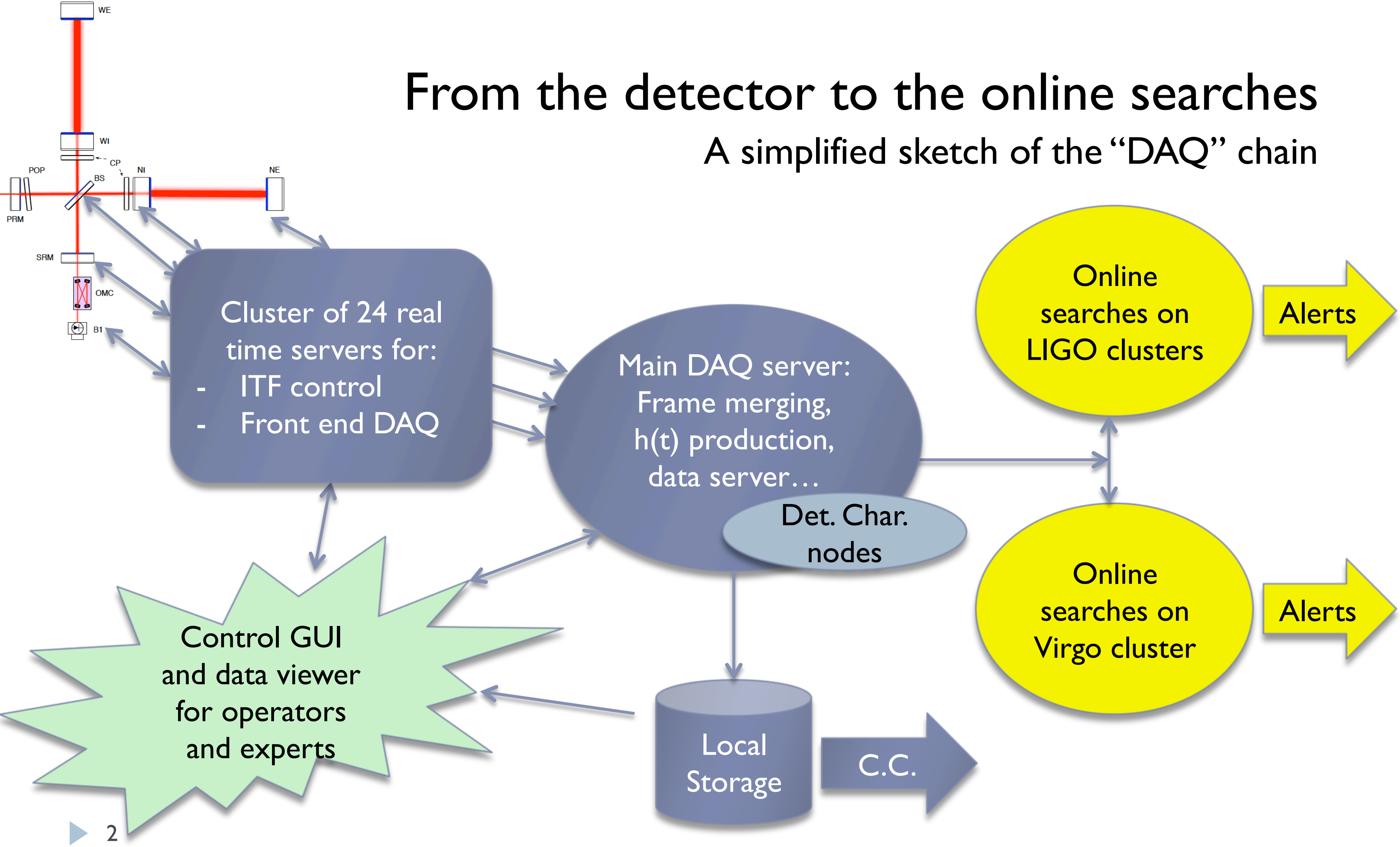
h(t) production and online searches

B. Mours (LAPP-Annecy)

July 2nd, 2018

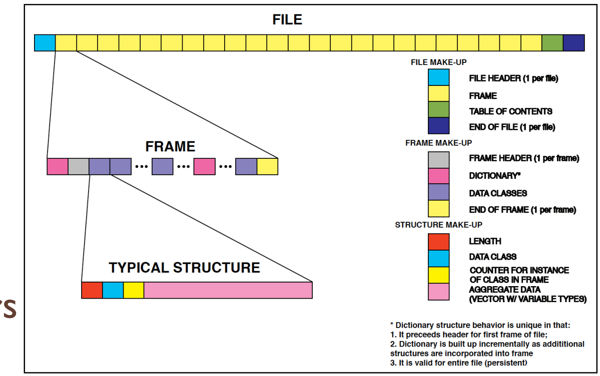
From the detector to the online searches

A simplified sketch of the "DAQ" chain



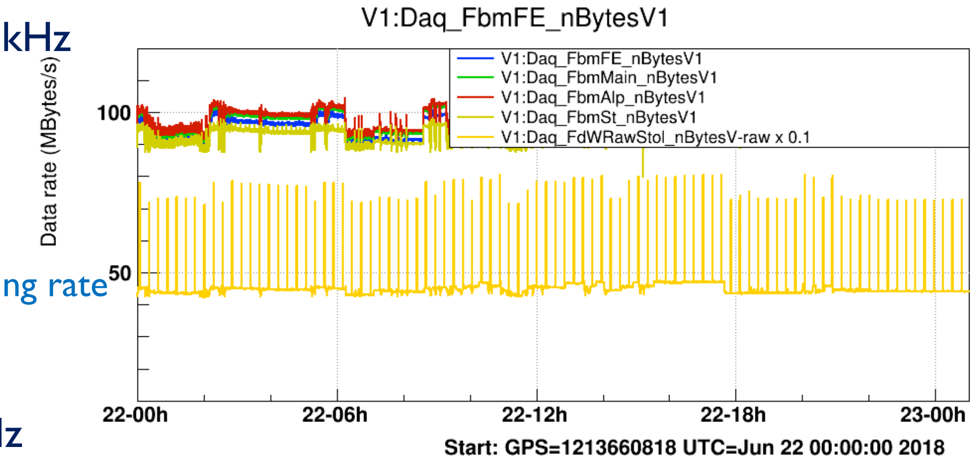
Virgo data and frame format

- Virgo is closer to an accelerator rather than a high energy detector:
  - The primary data stream stream is a continuous stream of channels not events
- Data are organized in frames
  - Frames are a time window (from 1s to hours), containing multiple parallel channels
  - A Virgo frame files
    - Could contains multiple frames
    - Has a Table Of Content for fast single channel access
    - Includes checksums for standalone file verification
    - Include lossless compression algorithms
    - The frame format is common to all (ground based) GW detectors
  - We have different types of frames, see later
  - Tools to merge frames, make them longer, shorter…
- Events are produced downstream from a small number of channels
  - Small data rate compared to the raw data stream



Main types of data produced by the Virgo DAQ

- Raw data: 32700 channels from 1 Hz to 100 kHz
  - + samples @ 400 MHz, images…
  - Archived at the computing centers
  - Now: around 50 MB/s 100 s. long files
- Full raw data
  - Raw data + debugging channels at higher sampling rate
  - Keep at the site on a circular buffer of few days
  - Rate = 80-90 MB/s 100 s. long files
- Reduced data stream: most channels at 50 Hz
  - Convenient for commissioning studies
  - Rate = 0.8 MB/s 43 files of 1.5 GB per day
- Trend data stream: min/mean/max/rms @ 1 Hz
  - Convenient for long term studies or the search
  - Rate = 0.06 MB/s 1 file of 4.6 GB per day to archive
- h(t) frames:
  - The useful stream for event searches
  - Rate = 0.06 MB/s 43 files archived of 120 MB per day

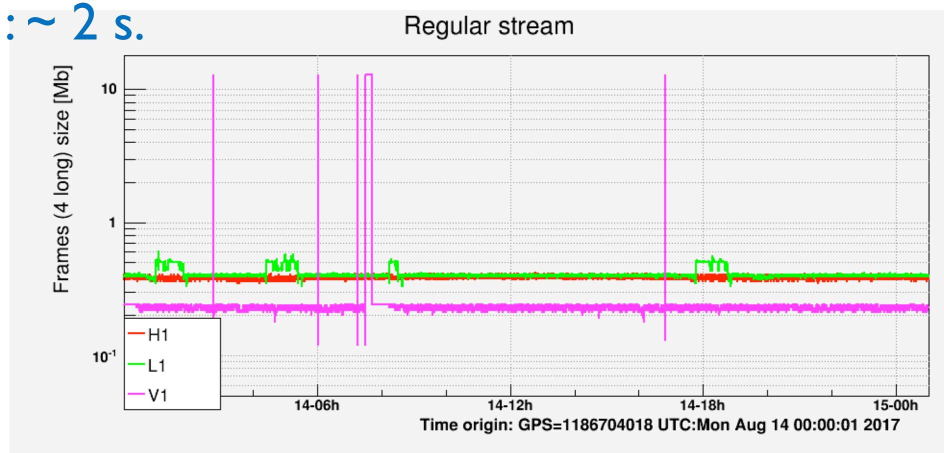


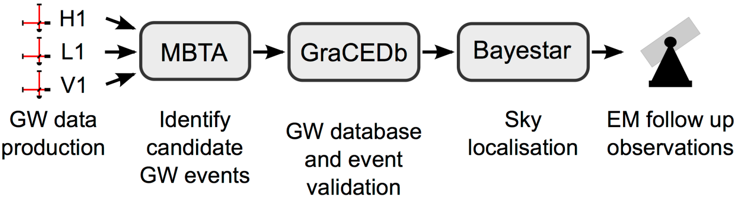
h(t) reconstruction

- Runs online to:
  - Provide direct feedback for the detector commissioning/operation
  - Feed online searches for real time alerts
    - Constrains: latency as low as possible: now 20 seconds; to be reduced by a factor 2 for O3.
- Algorithm:
  - Combined multiple input channels (PDs, mirror actuators, …) to compute h(t)
  - Light CPU resources needed
  - ***Reprocessing requires reading back the raw data***
- Embedded in the online "DAQ" chain
  - Uses the DAQ interfacing toolbox to transfer data (frames) between processes
  - Run on the main DAQ server: takes frames from shared memory
  - Output frames for:
    - "Raw" data stream storage
    - Detector characterization algorithms thought TCPIP and shared memory frame distribution
    - Online analysis: 4 seconds long low latency frames
    - Offline analysis: 2000 seconds long frames files

h(t) data exchange with LIGO

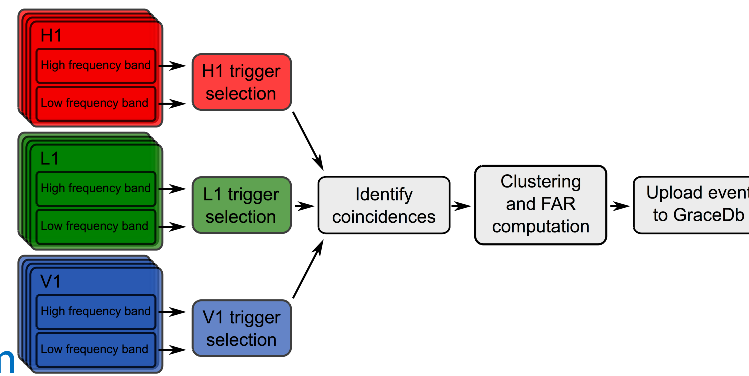
- Virgo h(t) is sent continuously to Caltech
  - Use Virgo DAQ tools for data exchange with LIGO
  - h(t) frames of 4 seconds long
- Receiving LIGO h(t) stream for online analysis running at Virgo
  - Low latency from the data transfer : ~ 2 s.
  - LIGO 4 s. long frame files: 0.4 MB
- Continuous exchange
  - Independent of the detector state
  - Low maintenance service
    - Few restart per year
- 2000 s. long file are imported by LIGO using LDR





Virgo CBC online search: MBTA

- Virgo developed pipeline to search for Compact Binary Coalescence (CBC)
- Run at Cascina since the first Virgo data taking
  - Was the first pipeline to search online for CBC events and send alert (S6-VSR3)
    - For O2/O3 one of the 3 low latency CBC pipelines
  - Use LIGO and Virgo data
  - Search for HL, HV, LV and HLV triggers
  - Uses the the DAQ tools for data IO
  - Alert produces in less than a minute
- Why running MBTA at the Virgo site?
  - Same software environment as the online system
  - Low latency data access for one of the detector
  - Redundancy with the CIT cluster
  - Pipeline could be monitored by the operators
    - Same control GUI as the Virgo main processes supervision



MBTA and ressources

- Template search on single detectors stream + coincidence step
  - Match filtering technic
  - About 200k templates
    - Use Multi Bands to optimize computation and reduce the number of "real" template used
  - Processing optimize to run on multicore machines to share real templates
    - Larger is the number of core, better is the load balancing for the triggers construction
- Fairly CPU effective
  - For O2: about 3 times more CPUs used for pycbc-live 3 and 10 times more for gstlal-online
  - 6 machines (two CPU with 8 physical cores) needed for a three detectors search in O2.
    - A total of 10 machines were available
      - Asked to double it: to run the production + injection stream pipeline + test pipeline
    - Disk space was a also limitation: not enough space to save intermediate products and tests results
      - Limited to 60 TB: asked to double it
- Remarks:
  - KAGRA may joint the end of O3 → more CPUs might be needed
  - Adding a (semi) coherent step might requires more CPUs
  - The shape of the sensitivity curve add some unknown (more and longer templates?)

A new pipeline for O3: CWB

- Coherent Wave Burst
  - Coherent search of "unmodeled" signals between detectors
- Already running at CIT and AEI for O2
- Study the feasibility of running it a Cascina for triggers with Virgo
  - Use Condor
  - Two machines for zero lag triggers
  - "Several" machines for background
  - On-going tests