Online Computing: Now vs Ideal

Franco Carbognani – EGO

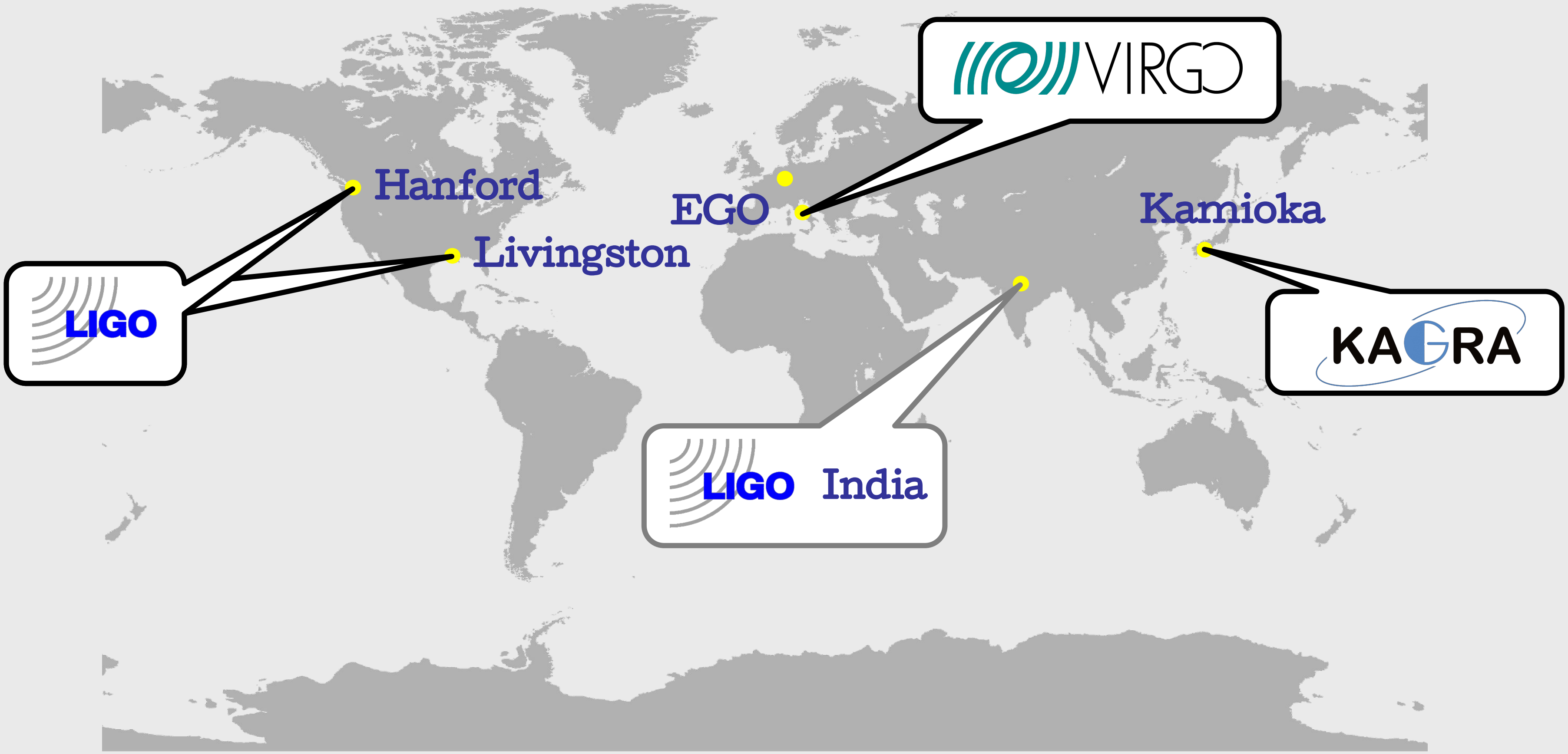
on behalf of the LIGO, Virgo and KAGRA collaborations

ET-PP Computing Workshop| 08 July 2024

Overview

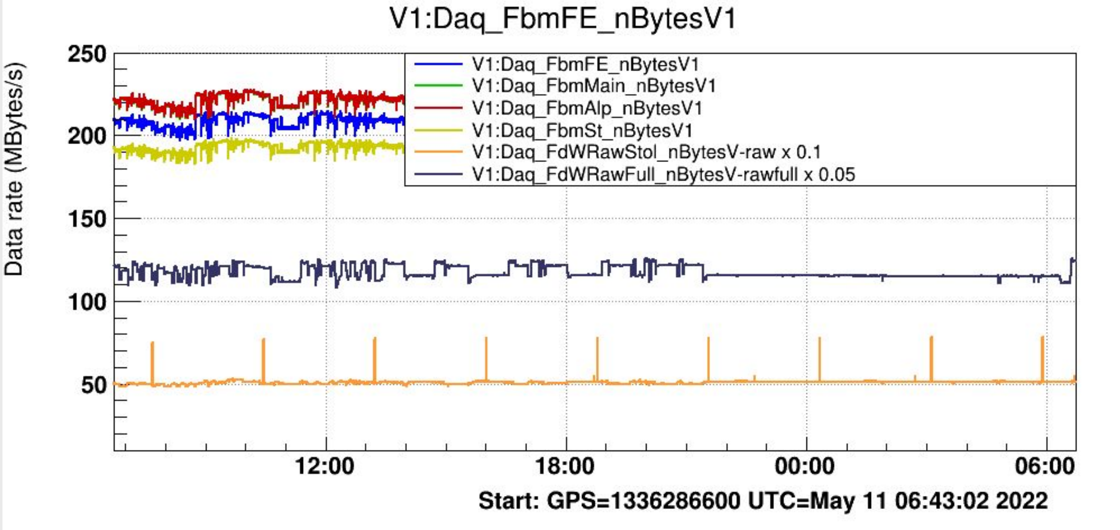
- A Worldwide Network
- Data and data flow schema
- Low-latency Analysis timescale
- The Challenge
- The Common Solutions: Now
- The Common Solutions: Ideal
- Conclusions

A Worldwide Network



Raw Data, ~1.5 PB/yr:

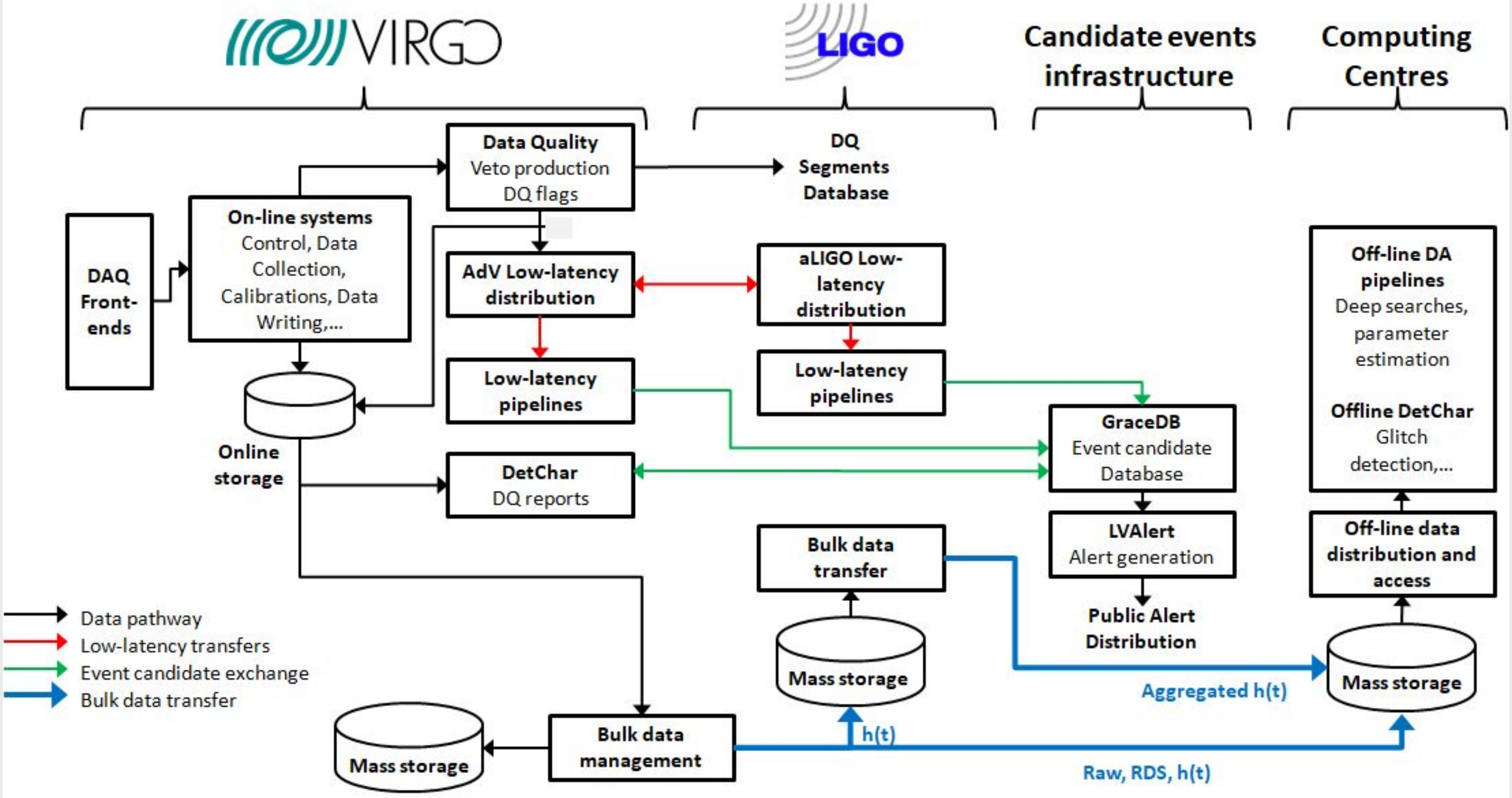
- **Full Bandwidth Raw**, not exported
- **Raw Data:** downsampled, include all auxiliary channels and calibrated/uncalibrated h(t) (~ 50MB/sec -> 4 TB/day -> 1.5 PB/yr), exported to Virgo Tier1 CC (CNAF, CC-IN2P3) for custodial storage during scientific runs
- **A few levels of reduced data sets** for various uses



Data for physics, ~ 5TB/yr/detector:

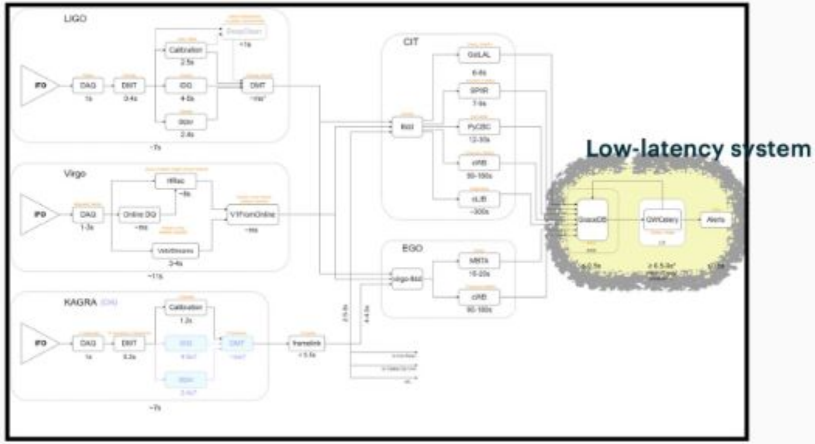
- **Virgo h(t): calibrated «strain» data**
  - sampled at 10 KHz, stored as ~1kSec frame files: aggregated h(t)
  - Includes state vector (data quality flags, vetoes,…)
- **LIGO (and KAGRA) h(t)**
  - Transferred online to EGO for low-latency searches and made available to IGWN CCs for offline analysis

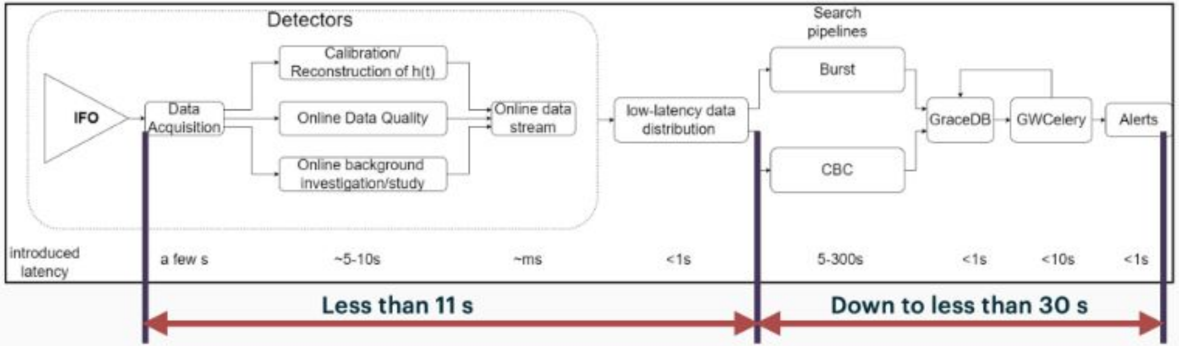
The Data Flow Schema



Low-latency Analysis timescale

- We are running extensive tests (already started - up to engineering runs) from data acquisitions (synthetic) to alert generation, and we are monitoring latency.
- We have the signal ready to be analyzed online in less than 11 seconds from the arrival of the (GW) signal at the detectors.
- That makes pre-merger alerts possible (with negative latency) and the first preliminary alerts in less than a minute.
- The study will also allow us to test the effectiveness of the online pipeline to detect and assess the properties of the signal.





End-to-end latency study group (https://wiki.ligo.org/Operations/O4EndToEndLatency)

9

roberto.depietri@unipr.it

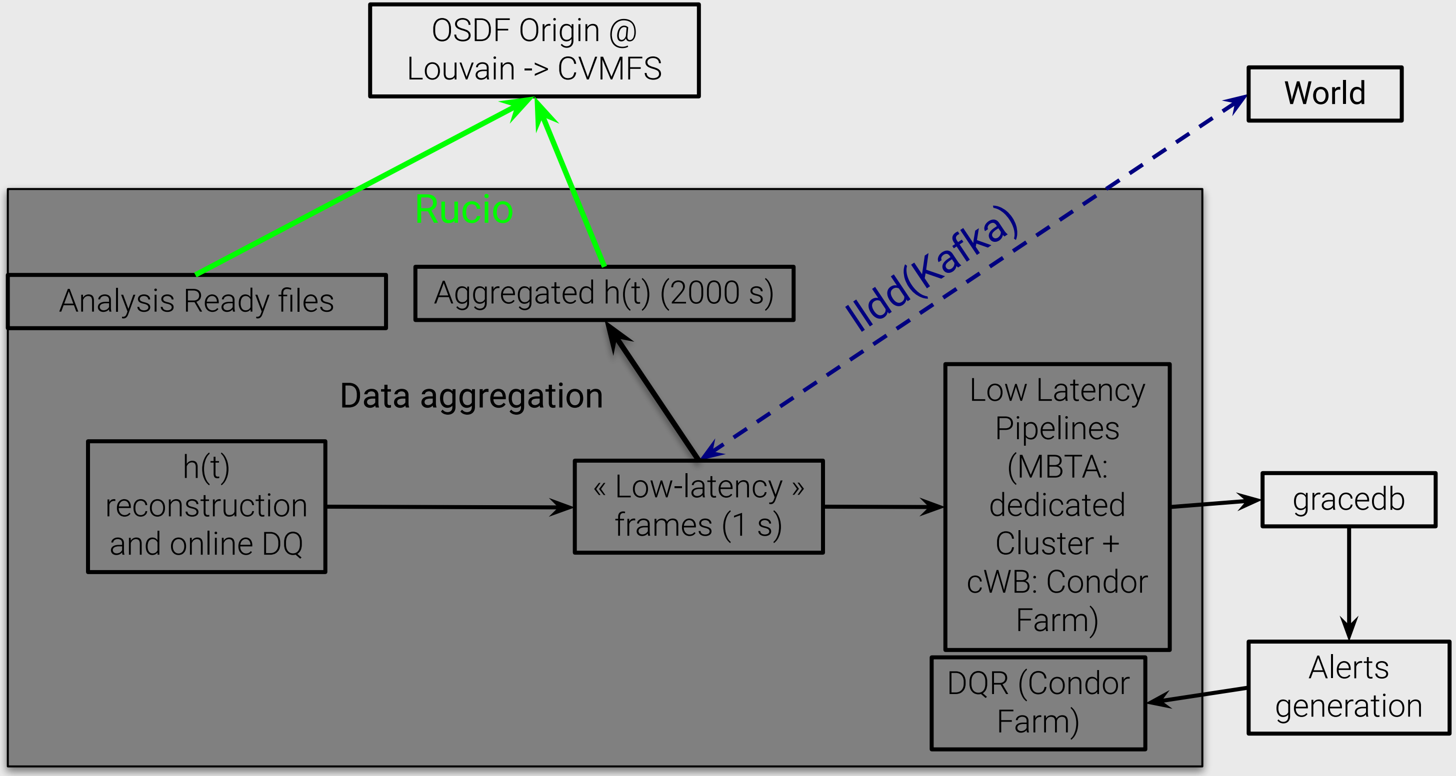
The (Online) Challenge

- The geographical separation of the detectors and the short timescale involved (e2e latency from signal detection to alerts generation in the order of 30 sec) imply the creation of a common distributed cyber infrastructure which must guarantee:
  - Adequate **storage and computing resources** for detector characterization, low-latency searches and alerts generation
  - **Low-latency data distribution** among the different observatories and computing clusters for low-latency searches
  - **Bulk** (Aggregated h(t) and raw) **data transfer**
  - An ubiquitous and uniform **running environment** on dedicated resources and heterogeneous infrastructures
  - A robust **support for development and operation**

The Common Solutions: Now

- Storage and computing resources
  - Dedicated or highly-prioritized resources provided by observatory computing centers
- Low-latency data distribution
  - Kafka + legacy solution (Cm)
- Running environment
  - IGWN environments: Conda + CVMFS
- Aggregated h(t) data transfer
  - Rucio
- Raw data transfer
  - Legacy Solution (based on iRODS and WebDAV)
- A robust support for development and operation
  - GitLab

Main components



The Common Solutions: Ideal

- Storage and computing resources
  - High Availability (HA) deployment (on top of Kubernetes)
- Low-latency data distribution
  - Kafka (fully mastering data distribution into large cluster of machines)
- Running environment
  - IGWN environments: Conda + CVMFS
- Aggregated h(t) transfer
  - Rucio
- Raw data transfer
  - Rucio
- A robust support for development and operation
  - GitLab

Storage and Computing Resources

Low-latency storage and computing mainly provided by observatory computing centers

- Low-latency alert infrastructure runs on dedicated resources with high priority to burst out into pool
- Search pipelines run on dedicated or highly-prioritised resources in an HTCondor-managed resource pool
- Fast, direct access to small data files / shared memory
- GraceDB production instance currently deployed in High Availability (HA) on AWS
- Alternative HA deployment via Kubernetes is being tested on INFN-Cloud at CNAF

High Availability (HA) deployment

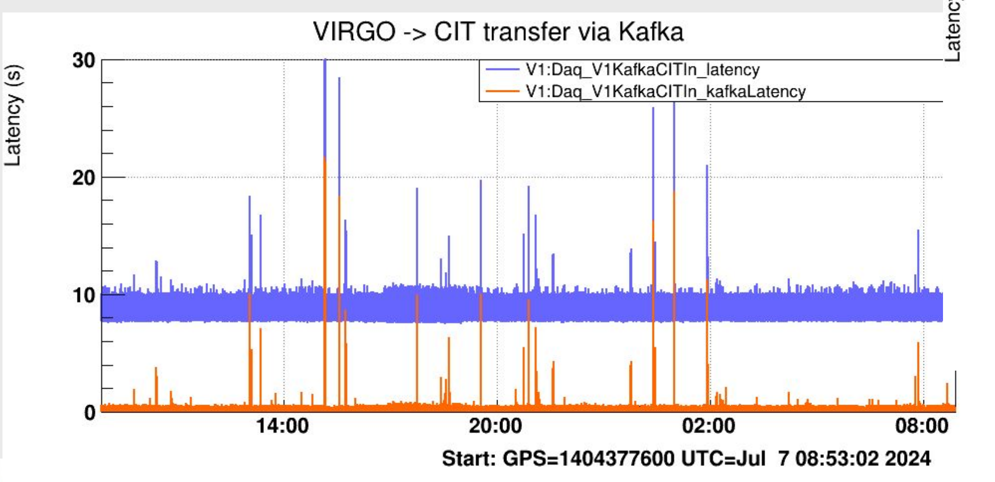
A research project is ongoing on a Kubernetes-based HA deployment for the alerts generation components (GraceDB, GWCelery, LVAlert) into the CNAF Computing Center Cloud.

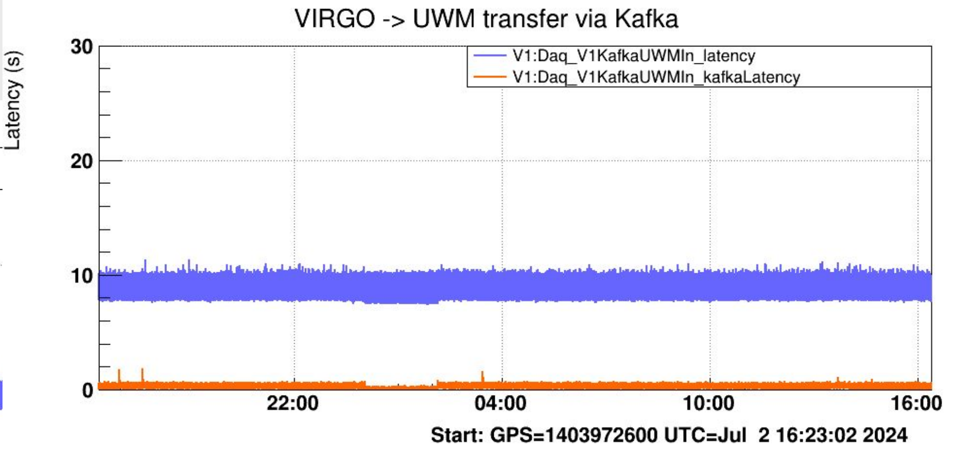
- To exploit support from CCs staff and achieve cloud provider independence
- Provide support for both high-availability deployment using K8S and on-demand HTCondor cluster using cloud-native tools
- A preliminary step to provide infrastructure to run low-latency searches off-site
- Not (yet) targeting a production deployment, possibly the test/playgroud tiers
- Progress has been slower than hoped…..

Low-latency data distribution: Kafka

- Kafka is a modern high throughput stream processing software
  - Built in redundancy
    - Can survive if stream stops or Kafka broker goes down
    - Replication so no data loss from service downs
  - Highly scalable and reconfigurable
    - Can easily add additional observatories data (done for KAGRA) and quickly reconfigure the network links
  - Production release of IGWN Low Latency Data Distribution software (lldd), build on top of Kafka and jointly developed by Ligo and Virgo, running smoothly in O4b.

- Currently experiencing some latency spikes
  - Traced down to be within the Caltec cluster (CIT) itself and not on the geographical links
  - Need to better master data distribution among cluster nodes via Kafka to proceed also in Cascina to deprecate the Cm based solution





Running Environment: IGWN Conda

Sustainable software development and distribution = **CVMFS** + **Conda**

Conda environments hosted in CVMFS file system available at all IGWN sites and CC

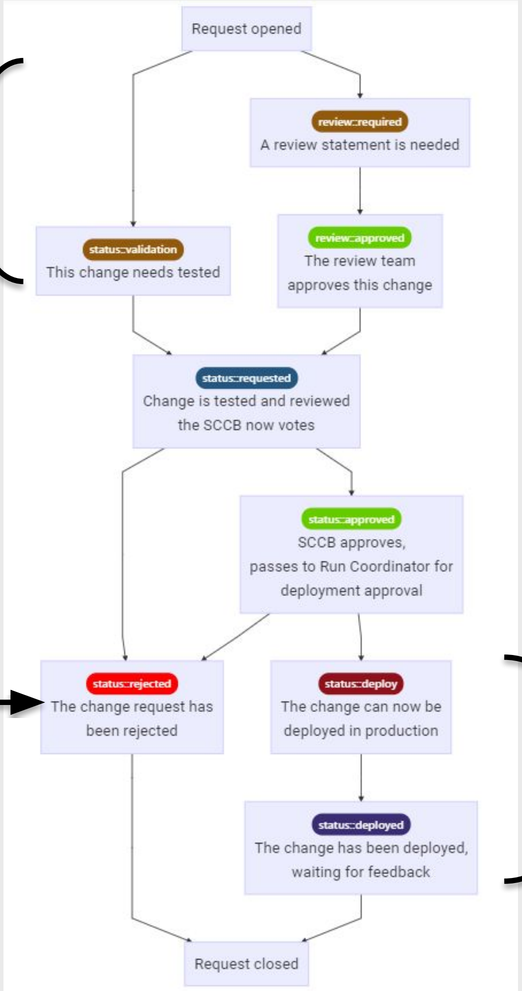
Running Environment: IGWN Conda

- IGWN Conda Distribution provides pre-built, automatically-distributed environments of approved software
- available via CVMFS on any machine (no authentication required)
- can be replicated on any workstation
- Provide effective OS independence
- Leverage on CMake/Meson for software build
- Provide a very effective solution for otherwise unmaintainable number of custom software builds
- On large part of IGWN sites (including Virgo Cascina) the IGWN environment can be activated manually or is by default at user login

Software deployment into IGWN Conda Distribution

Includes:
- **Packaging**
- **Testing**
- **Scientific review (with the package developer)**

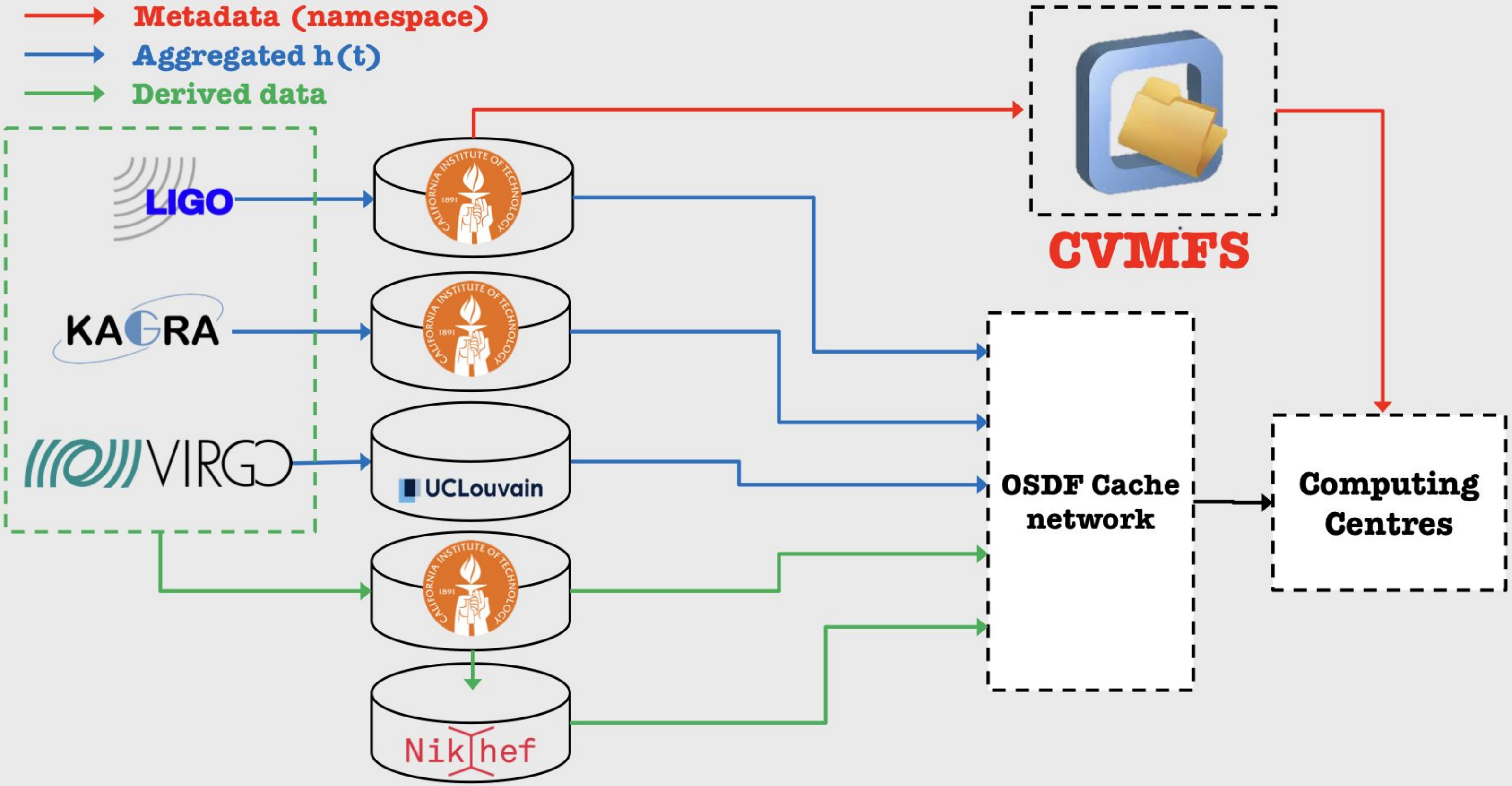
Very rarely happens. Typically, only done if the person requesting the package no longer wants it in the IGWN Conda Dist.



- Software deployment is controlled via change control board (SCCB)
- Software must be approved by SCCB and the relevant scientific review committee before being used for production analysis

This stage is done by a Run coordinator

Aggregated h(t) data transfer (O4 OSDF configuration)



Aggregated h(t) data transfer

Rucio for Virgo

- Virgo data transfers:
  - Managed by Rucio: Aggregated H(t), EGO -> Louvain -> CVMFS
  - Planned to be managed by Rucio: Environmental data (magnetometers), EGO -> Louvain -> CVMFS -> available at CIT
  - For custodial storage (not with rucio): Raw + Aggregated, EGO -> CCIN2P3 and CNAF:
    - Custom transfer framework using gfal and iRODs
  - Low-latency transfers: Kafka based

Next steps:

- During O4:
  - Consolidate transfer infrastructure based on Rucio
- After O4:
  - test and possibly use Rucio also for raw data to CCIN2P3 and CNAF

5

VIR-1074A-22 & LIGO-G2200009-v1

Federica.Legger@to.infn.it

Aggregated h(t) data transfer

- Aggregated 2000 sec h(t) data (aka online files) have been flowing from Cascina to the OSDF/CVMFS Louvain origin using Rucio since before the start of O4a
- Same mechanisms will be used for the distribution of the offline Analysis Ready files and a test will be soon setup using a set of renamed aggregated h(t) files

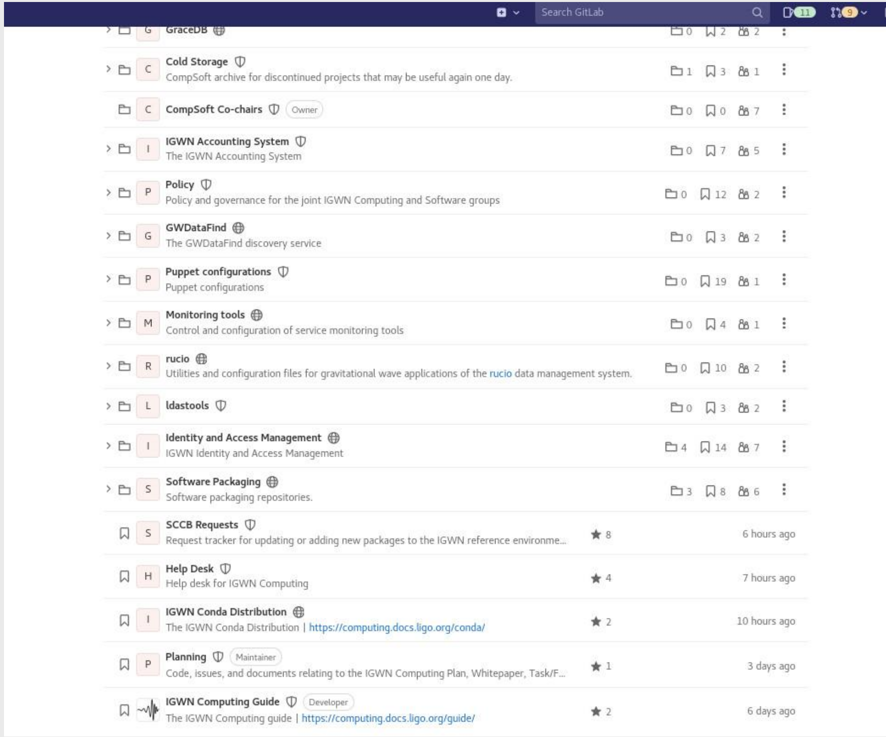
Raw data transfer

- The iRODS + WebDAV Raw data transfer to CCIN2P3 and CNAF is going well and data is flowing into dedicated O4b directories
- Some problems have been experienced on the iRODS to HPSS “translation”
- Preliminary tests on Rucio for raw data transfer have started
  - Infrastructure already setup
  - For the first test, data will be transferred from Cascina to Torino

Development and Operation

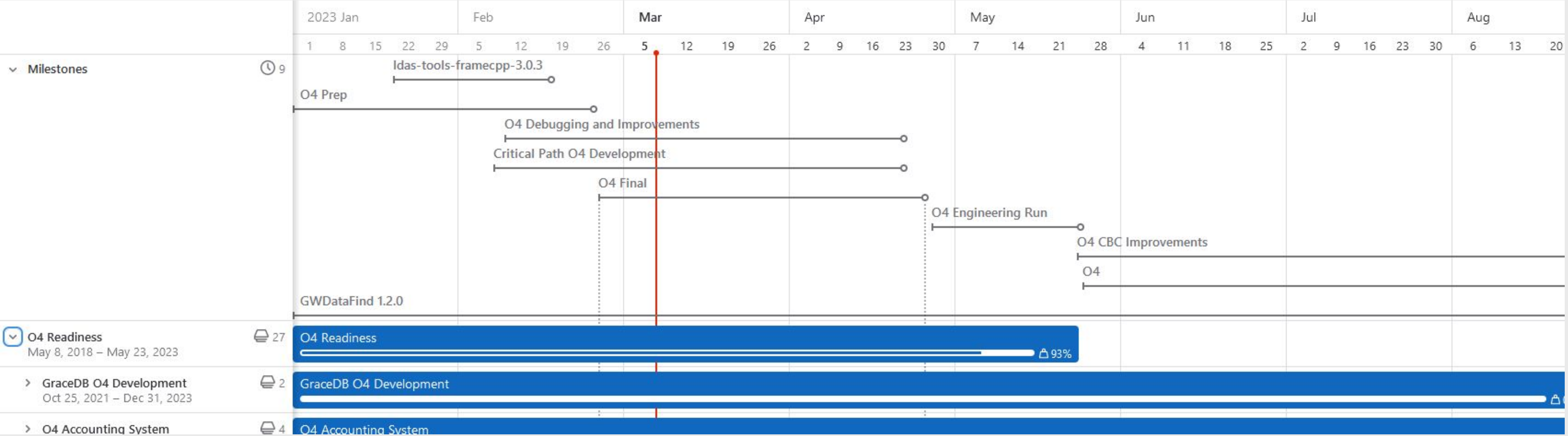
LIGO, Virgo and KAGRA Computing & Software teams increasingly working together using GitLab centered DevOps practices. IGWN Gitlab provides/supports:

- Code repository for IGWN Detectors Control, Data Analysis and Low-latency software
  - Enables easy collaboration on software
- Continuous integration (CI) capabilities to enable the automation of building, testing and deployment of code.
- Software Configuration Control Board (SCCB) activities
- IGWN Computing HelpDesk (via tickets)



Development and Operation

- Conda software distribution management
- IGWN Computing Guide (via GitLab Pages)
- IGWN Computing Planning (via the growing GitLab support for projects management)
  - Issues
  - Epics



Conclusions

- The transition from custom, incompatible LIGO, Virgo and KAGRA-specific infrastructures to a common multi-collaboration infrastructure, based on mainstream tools, is well ongoing and has payed already many dividends (e.g. Gitlab, Conda, Rucio, ..)
- The upgrades implemented in time for the O4b Science Run are serving us well
- We need is to steadily proceed within the identified common solutions path