

# Virgo and Gravitational-Wave Computing in Europe

Stefano Bagnasco<sup>1,\*</sup> for the Virgo Collaboration and the LIGO scientific collaboration.

<sup>1</sup>Istituto Nazionale di Fisica Nucleare, Sezione di Torino, 10125 Torino, Italy

**Abstract.** Advanced Virgo is an interferometer for the detection of gravitational waves at the European Gravitational Observatory in Italy. Along with the two Advanced LIGO interferometers in the US, Advanced Virgo is being used to collect data from astrophysical sources such as compact binary coalescences and is currently running the third observational period, collecting gravitational wave event candidates at a rate of more than once per week. Data from the interferometer are processed by running search pipelines for several expected signals, from coalescing compact binaries to continuous waves and burst events. Furthermore, detector characterisation studies are run. Some of the processing needs to be done with low latency, to be able to provide triggers for other observatories and make multi-messenger observations possible. Deep searches are run offline on external computing centres. Thus, data needs also to be reliably and promptly distributed from the EGO site to computer centres in Europe and the US for further analysis and archival storage. Two of the defining characteristics of Virgo computing are the heterogeneity of the activities and the need to interoperate with LIGO. A very wide array of analysis pipelines differing in scientific target, implementation details and running environment assumptions have to be allowed to run ubiquitously and uniformly on dedicated resources and, in perspective, on heterogeneous infrastructures. The current status, possible strategies and outlook of Virgo computing are discussed.

## 1 Introduction

Gravitational-Wave (GW) Astronomy was born on September 14, 2015, when two detectors comprising the US Laser Interferometer Gravitational-Wave Observatory (LIGO) observed GW from a pair of colliding black holes [1]. Then, the joint discovery of a merging binary neutron star by Advanced LIGO (aLIGO) and Advanced Virgo (AdV) on August 17, 2017 triggered an electromagnetic follow-up of the source. An electromagnetic counterpart was actually observed by several land- and satellite-based observatories: that was the first occurrence of a new type of multimessenger astronomy [2]. Since then the aLIGO and AdV interferometers have detected many more GW event candidates, at a nearly weekly rate during the last observation period (lasting from May 2019 to April 2020).

---

\* Corresponding author: [stefano.bagnasco@to.infn.it](mailto:stefano.bagnasco@to.infn.it)

An upgrade of the AdV interferometer, the AdV+ program, will boost the sensitivity approximately by a factor of 2 in O4, starting in 2021, and by a factor of 3 to 5 in O5, starting in 2024 [3]. The expected event rate scales with the cube of the sensitivity improvement, so the Collaboration is enhancing and consolidating its computing infrastructure to be able to cope with the increased processing demands.

## 2 Advanced Virgo data processing

For the purpose of this discussion, AdV computing activities can be roughly split into three categories: online (detector control and monitoring, data acquisition, etc.), low-latency searches (that generate triggers for multimessenger follow-up) and offline data analysis, both for detector commissioning and characterisation and scientific analysis (offline searches and parameter estimation). Even though some comment will be made about low-latency searches, the following discussion will mostly focus on offline computing. Given this scope, the data acquisition and online processing description we are giving below is necessarily very sketchy.

GW research is a collaborative effort: the usefulness of GW data is vastly enhanced by combining the observations from several interferometers. For example, the stochastic GW background is expected to be below the instrumental noise of a single detector, and sky localisation takes great advantage from multiple observations of the same event. Thus, AdV and aLIGO co-ordinately pursue a common science program through common data analysis campaigns, sharing data and computing resources. The KAGRA collaboration, operating an interferometer in Japan that will start science runs in 2020, is also joining the effort in the LVK collaboration.

### 2.1 Data processing workflow

GW data have a relatively simple structure. In the format generally used for signal searches and parameter estimation, they reduce to a time series of the dimensionless “strain” variable  $h(t)$ , sampled at 16.384 kHz, with an associated “state vector” of detector status information and data quality flags. The total amount of such data is relatively small, about 5TB per year per interferometer. AdV also produces about 1PB of raw data per year, that is exported to two external sites (CNAF in Bologna, Italy and CC-IN2P3 in Lyon, France) for custodial storage.

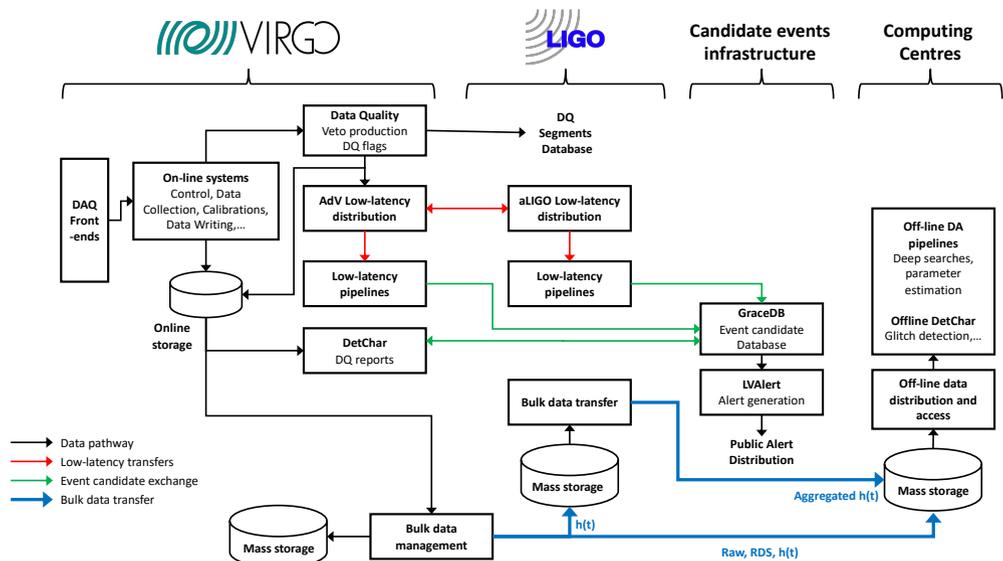
During science runs, data coming from the interferometer are processed in real-time on the online computing infrastructure in Cascina, Italy, formatted and written to disk for subsequent analysis; some online analysis of the data is also performed, while the data still resides in shared memory on dedicated servers, mostly for detector monitoring purposes.

Data coming out of the interferometer need to be analysed with the lowest possible latency to be able to timely deliver triggers for multi-messenger follow-up observations,  $h(t)$ . AdV low-latency data are transferred in quasi-real-time to aLIGO computing centres and the other way around so that low-latency searches can be performed on data from all the three interferometers.

Low-latency search pipelines run on the Cascina infrastructure (and symmetrically in US centres) to search for transient signals, Compact Binary Coalescences (CBC) and unmodeled burst signals. GW event candidates detected by the low-latency analysis pipelines are submitted to the GraceDB database [4]. A mostly automated system is then used to generate Public Alerts, that are distributed through NASA’s Gamma-ray Coordinates Network [5].

Offline search pipelines are then run on saved data. These include searches difficult to perform in low-latency mode, such as those for continuous signals, and deeper searches for burst signals that may have been missed. Furthermore, for each GW event candidate Parameter Estimation analyses are run to complete the information about the candidate. Lastly, some Detector Characterisation-related activity is also run offline, like the Omicron glitch-detection pipeline. Offline computing activities take place essentially off-site, in the two main Virgo computing centres (CNAF and CC-IN2P3) in Europe, on LIGO Data Grid (LDG) resources in the US and increasingly on a more distributed infrastructure discussed below. Data for offline analyses are distributed to external computing centres as “aggregated  $h(t)$ ”, again including data from all three interferometers.

A schema of the data flow, including communication between the two collaborations, is shown in Fig. 1.



**Fig. 1.** AdV dataflow schema.

## 2.2 The distributed computing infrastructure and its evolution

As mentioned above, offline computing activities in Europe used to be performed essentially at the two main AdV computing centres (CNAF and CC-IN2P3) where custodial storage for raw data is also located. Lately, several other centres from collaborating institutes have been joining the effort, providing computing resources to the collaboration: Nikhef in Amsterdam, the Netherlands; PIC in Barcelona, Spain; the Université Catholique de Louvain in Louvain-la-Neuve, Belgium and Cyfronet in Krakow, Poland, and possibly more will come in the future.

Alongside an expanding computing infrastructure, also some other considerations required a rethinking of the overall computing plans.

First, several currently used implementations of the required functionalities (see below for details) are often based on legacy architectures, old and possibly no longer maintained

tools (like for example CMT [6] and SVN [7] for software management) or on custom solutions, for which management and maintenance take a heavy toll on limited computing expert manpower.

Then, for mostly historical reasons, GW analysis pipelines are very heterogeneous in implementation choices (being e.g. written in Matlab, Python, C++) and often tightly coupled to a specific running environment (such as, for example, submission to a specific infrastructure with dependencies in the code on the particular batch system flavour and local configuration), each having been developed independently and without a common framework. As a consequence of this, there is no clear boundary between «middleware» and the analysis code proper.

Finally, full interoperability with the US-based infrastructure is mandatory.

At the same time, the aLIGO collaboration was facing similar challenges. For example, needing to expand the GW computing infrastructure in the US away from the proprietary, uniform Ligo Data Grid platform towards a more distributed grid computing infrastructure on shared resources, and obtained strong support from the Open Science Grid (OSG) that operates the scientific computing Grid in the US.

Thus, a plan has been devised by the two collaborations to progress from two separate computing infrastructures towards maximum interoperability with a common architecture, finally aiming at a fully uniform common platform, that will also be shared with KAGRA.

As a common container for the initiative, the International Gravitational-Wave observatories Network (IGWN) has been created, first as an informal coordination effort between the collaborations and a common namespace for tools and services, then possibly evolving in a more formal agreement. In the following, we will refer to the system being currently deployed as the “IGWN architecture”.

### 3 The IGWN architecture

The following is a bird’s eye view of the IGWN architecture at its current development status. As mentioned above, in the following description we will focus on the aspects more strictly pertaining to offline distributed computing; so, for example, we will not discuss low-latency data exchange.

#### 3.1 Functionalities and implementation guidelines

The computing requirements for Gravitational-Wave data processing are not structurally different from those of, for example, High Energy Physics experiments. The main needed functionalities are, as can be seen, quite similar:

- **Bulk data transfer and storage:** safely and efficiently transfer all data to custodial storage in computing centres;
- **Software packaging and distribution:** manage software lifecycle, and make pipeline packages available ubiquitously;
- **Data distribution:** make  $h(t)$  (and possibly other long-lived data) available to worker nodes in computing centres anywhere, and possibly also to single workstations;
- **Data cataloguing and bookkeeping:** organise all data and metadata and provide querying and discovering capabilities;
- **Job lifecycle management:** provide a uniform job submission and runtime environment;

- **High-level workload management:** keep a database of all jobs and allow the enforcement of priorities and scheduling strategies;
- **Monitoring and accounting:** monitor distributed computing, checking performance and looking for issues, and provide reliable accounting both at the user/job and site level;
- **Authentication, Authorisation and Identity management:** provide consistent AAI across computing infrastructures.

The guidelines for implementation of these functionalities in the IGWN architecture directly derive from the requirements of overcoming current limitations and define and deploy a common and sustainable GW computing environment. Thus:

- Provide a uniform and simple to use runtime environment for offline pipelines;
- Guarantee full interoperability between Virgo, LIGO (and KAGRA) pipelines and infrastructures;
- Ensure future scalability and the opportunity to exploit heterogeneous resources
- Adopt the smallest possible set of mainstream, widely used tools, leveraging upon HEP experience.

## 3.2 Tools and implementation

### 3.2.1 Software management and distribution

As mentioned above, software management was based on legacy tools and “manual” distribution of software packages to computing sites.

In the IGWN architecture, most software development goes through a common GitLab [8] instance with Continuous Integration features. Pipeline software and its dependencies are packaged as Conda [9] environments, that are then distributed to centres through a CVMFS repository, managed by OSG. CVMFS [10] is a tool originally developed at CERN to distribute scientific software to computing centres via a hierarchy of Squid proxies. CVMFS provides a read-only filesystem that can be mounted on worker nodes (and personal workstations as well), is very stable and scalable, and widely used in the HEP computing community.

The environments can also be packaged as Docker [11] or Singularity [12] containers, again distributed via the CVMFS repository.

A separate CVMFS repository is currently managed by EGO in Cascina, to ease the transition phase in preparation for a common software distribution.

### 3.2.2 Data management

The existing data management infrastructure relies mostly on the LIGO Data Replicator (LDR) [13] and several custom tools; there is no centralised file catalogue. Available data can be discovered by querying a site’s specific service through *gw\_data\_find*, a tool developed by LIGO which relies on file metadata (e.g. observatory, GPS time interval and such) being encoded in the filename. Alternatively, Virgo users use a minimal file catalogue functionality through FFL (“Frame File List”) files, that associate data files with their GPS time interval.

The tool of choice for bulk data transfer in IGWN is Rucio [14], a widely used system for scientific data management originally developed in the context of the ATLAS collaboration. Rucio is also a promising candidate for many other data management needs. In its first step of adoption, Rucio will (and in some cases already does) manage all scheduled data transfers between computing centres, with the exclusion of low-latency links managed by Apache Kafka [15].

Besides scheduled data transfers,  $h(t)$  data is being made available to computing centres in the US through StashCache, the OSG Data Federation tool [16]. A reference authoritative copy of the data is kept in a StashCache “origin” at the University of Nebraska, and cache instances are deployed according to needs close to client computing centres. In Europe, a cache instance is for example deployed for general use (i.e. not limited to the LIGO/Virgo collaboration) at the University of Amsterdam, and another one was deployed at CNAF specifically for LIGO/Virgo use. The exact topology of cache instances in Europe is not yet defined and will depend upon performance test results across all the involved computing centres.

A datafile namespace is exported to sites via another CVMFS repository, that can be browsed and searched; actual data bytes are then read from the closest StashCache instance through the *externalData* feature.

CVMFS itself is designed to distribute open-source code and provides no authorization mechanism, so the namespace is public. However, x.509-based authentication is provided for file access by an external OSG-developed plugin.

A dedicated instance of the *gw\_data\_find* server scans the CVMFS namespace to enable researchers to discover data distributed via CVMFS and StashCache.

Rucio’s file catalogue features allow for many possible developments to streamline the system and provide additional functionalities. For example, a possible upgrade that does not break the *gw\_data\_find* semantics but allows many more functionalities is to use the Rucio catalogue as a backend, reimplementing it to use its APIs instead of the server. A more speculative investigation is ongoing on the possibility of mounting a filesystem-like view of the Rucio catalogue database: this could be exploited to replace CVMFS and provide the same browsing functionality without disrupting the existing user workflows.

### 3.2.3 Workload management

Currently, GW analysis offline jobs are run in a variety of ways, from plain batch job submission on dedicated computing clusters to submission to Grid infrastructures.

In order to provide a uniform running environment (and if possible not only uniform job submission and management semantics but also syntax) the tool of choice for workload management is HTCondor [17], a very widely used high-throughput-specific workload management system that comes with a huge array of features. Once an HTCondor submit file is written to describe a given job (or even, through DAGs, a full workflow) it can be used with very little change to exploit different possible computing infrastructures, for example:

- a local or remote HTCondor cluster, possibly with the opportunity of off-loading to other clusters via the “flocking” mechanism;
- a remote HTCondor-CE [18], a flavour of Grid Computing Element based on HTCondor being widely adopted in high-throughput computing centres worldwide (with CNAF being a very relevant instance);
- a full Grid infrastructure, through dedicated “submit nodes” and the GlideInWMS [19] architecture.

We plan the latter to be, in due time, the main way of providing computing resources to the collaborations, at least for “production” activities; submit nodes are deployed and operated in the US by LIGO, and a first one in Europe is being commissioned at Nikhef. Having a limited number of control points as “pilot factory” instances allows the enforcement of priorities and the easy gathering of accounting information. However, this does not preclude the possibility to easily run also on other types of infrastructure, and of course allows for a gradual transition from local running to fully Grid-like distributed computing. One more opportunity is the capability of deploying on-demand HTCondor clusters on Cloud resources through for example DoDAS [20], condor\_annex (a mechanism provided by HTCondor) or other mechanisms, which is being explored. This would allow the collaborations to exploit heterogeneous or opportunistic resources transparently.

## 4 Conclusions

Gravitational-Wave computing, in Europe and elsewhere, is making a significant step forward towards organized computing. The IGWN effort means a coordinated and common infrastructure between the AdV, aLIGO and KAGRA collaborations, and a uniform running environment across different resource providers.

The adoption of common mainstream solutions like HTCondor or Rucio means not only a reduced management and maintenance burden but also that the Gravitational-Wave collaborations are joining the larger physics community in the common effort towards next generation experiments’ computing challenges.

Even though several details of the IGWN architecture, and many practical deployment details, are not yet final, a step-by-step implementation and testing activity is going on, and significant effort is also being dedicated to the nontrivial task of porting complex analysis pipeline software to the new running environment.

## References

1. B. P. Abbott *et al.* (LIGO Scientific Collaboration, Virgo Collaboration), *Phys. Rev. Lett.* **116** 061102 (2016)
2. B. P. Abbott *et al.* (LIGO Scientific Collaboration, Virgo Collaboration, Fermi GBM, INTEGRAL, IceCube Collaboration, AstroSat Cadmium Zinc Telluride Imager Team, IPN Collaboration, Insight-HXMT Collaboration, ANTARES Collaboration, Swift Collaboration, AGILE Team, 1M2H Team, the Dark Energy Camera GW-EM Collaboration and the DES Collaboration, DLT40 Collaboration, GRAWITA: GRAvitational Wave Inaf TeAm, Fermi Large Area Telescope Collaboration, ATCA: Australia Telescope Compact Array, Las Cumbres Observatory Group, OzGrav, DWF (Deeper, Wider, Faster program), AST3, and CAASTRO Collaborations, VINROUGE Collaboration, MASTER Collaboration, J-GEM, GROWTH, JAGWAR, CaltechNRAO, TTU-NRAO, and NuSTAR Collaborations, Pan-STARRS, MAXI Team, TZAC Consortium, KU Collaboration, Nordic Optical Telescope, ePESSTO, GROND, Texas Tech University, SALT Group, TOROS: Transient Robotic Observatory of the South Collaboration, BOOTES Collaboration, MWA: Muchison Widefield Array, CALET Collaboration, IKI-GW Follow-up Collaboration, H.E.S.S. Collaboration, LOFAR Collaboration, LWA: Long Wavelength Array, HAWC Collaboration, Pierre Auger Collaboration, ALMA Collaboration, Euro VLBI Team, Pi of the Sky Collaboration, Chandra Team at McGill University, DFN: Desert Fireball Network, ATLAS, High Time Resolution Universe Survey, RIMAS and RATIR, SKA South Africa/MeerKAT), *Astrophys. J. Lett.* **848** L12 (2017)

3. B. P. Abbott *et al.* (KAGRA Collaboration, LIGO Scientific Collaboration, and Virgo Collaboration), arXiv:1304.0670v10 (2020)
4. LIGO Scientific Collaboration, Virgo Collaboration, “GraceDB” [Online], 2019. Available at <https://gracedb.ligo.org> [Accessed: January 16, 2020]
5. “NASA Gamma-ray Coordinates Network” [Online]. Available at <https://gcn.gsfc.nasa.gov> [Accessed: January 16, 2020]
6. C. Arnault, in *Proceedings of CHEP 2000*, Padova, Italy (2000)
7. The Apache Software Foundation, “Subversion” [Software], 2019. Available from <https://subversion.apache.org> [Accessed: January 16, 2020]
8. The GitLab project, “GitLab” [Software], 2019. Available from <https://about.gitlab.com> [Accessed: January 16, 2020]
9. The Anaconda Project, “Conda” [Software], 2019. Available from <https://github.com/conda/conda> [Accessed: January 16, 2020]
10. J. Blomer *et al.*, J Phys: Conf Ser **331** 042003 (2011)
11. Docker Inc., “Docker” [Software], 2019. Available from <https://www.docker.com> [Accessed: January 16, 2020]
12. Sylabs.io, “Singularity” [Software], 2019. Available from <https://sylabs.io/docs> [Accessed: January 16, 2020]
13. A. Chervenak, R. Schuler, C. Kesselman, S. Koranda and B. Moe, in *proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, Seattle, WA, USA (2005)
14. M. Barisits *et al.*, Comput Softw Big Sci **3** 11 (2019)
15. The Apache Software Foundation, “Kafka” [Software], 2019. Available from <https://kafka.apache.org> [Accessed: January 16, 2020]
16. D. Weitzel *et al.*, in *Proceedings of the Practice and Experience on Advanced Research Computing*, Chicago, IL, USA (2019).
17. D. Thain, T. Tannenbaum, and M. Livny, Concurr Comp-Pract E **17** 323-356 (2005)
18. B. Bockelman *et al.*, J Phys: Conf Series **664** 062003 (2015)
19. I. Sfiligoi *et al.*, in *Proceeding of the 2009 WRI World Congress on Computer Science and Information Engineering*, Los Angeles, CA, USA (2009)
20. D. Spiga *et al.*, EPJ Web Conf **214** 07027 (2019)