CNRS
Centre National de la Recherche Scientifique

INFN
Istituto Nazionale di Fisica Nucleare

The AdV Computing Model. Version 1.0.
VIR-0129D-13

The Virgo collaboration and the Ego IT Department

Date : October 12, 2013

Abstract: We present here the Advanced Virgo Computing Model, which we have defined to guarantee a production and analysis system which gives an easy and robust access to data and resources, for both commissioning and analysis. This document is intended to be a living document, updated with well defined cycles. In fact, an “Implementation Plan” will then describe the technical solutions, as they are foreseen with the actual computing resources, together with plans for testing them. The Model is sustained by a “Management Plan”, which addresses the management procedures to make reality checks on it. The Model is composed by five parts: Workflows, Data Model, Data Management Distribution and Access, Software description and Computing Facilities resource requirements.

VIRGO Collaboration
EGO - Via E. Amaldi - I-56021 S. Stefano a Macerata, Cascina (Pisa)
Secretariat: Telephone (39) 050 752 511 - FAX (39) 050 752 550 - e-mail perus@ego-gw.it

Contents

Executive Summary

The Computing Model (CM) for Advanced Virgo (AdV) described here has been written taking advantage of the experience gained so far with the data taking and analysis from the first engineering runs to the latest Science runs VSR1-VSR4 (the last run ended in September 2011). It also takes into account the technological progresses of these years, from the original Virgo Plan, which is dated back to the year 2002, VIR-PLA-DIR-7000-122 [2]. The fundamental focus of the CM is to collect the requirements of the science data analysis (DA) groups and to find optimal solutions to fulfill them.

The CM also reflects needs and constraints arised from the LIGO/Virgo agreement [3], which we have been facing during the last years, and have finally been able to address in an organized way in this CM.

This is intended to be a living document, following a regular schedule. A natural schedule could be of the order of six months. During the first period of transition towards "Advanced Detectors Era" (ADE), it can be given by the Engineering Runs, a sequence of test runs scheduled by our LIGO colleagues, from January 2012 up to Dec. 2014, which we are using to exercize our DA software and organization towards ADE. Obviously, there are projects for which this will not be enough and will need to be finally tested and refined on real data, through simulations done injecting signals into real or simulated data (Mock Data Challenges, MDC). Being a "Computing Model" it does not contain detailed technical, manpower and costs information, which are discussed in the "Implementation Plan", which is again a living document, revised as we gain experience on its different items. The two documents will be jointly updated, since any result derived by new proposed technical solutions (described in the Implementation Plan) will lead to a modification of the CM. A "Management Plan" document closes the loop, detailing the procedure to be followed to verify the evolution of the CM and its implementation. The "Management Plan" is a very important (roughly) 2 pages document.

Gravitational wave (g.w.) searches run on real aLIGO-AdV data, and during MDC also on simulated data.

One of the main goals of the CM is to define a production and analysis system able to guarantee an access to data and resources transparent to the end users.

AdV has a hierarchical model for data production and distribution: different kinds of data are produced by the detector and firstly stored at the EGO site in Cascina. There is no permanent data storage in Cascina; we foresee to install a disk buffer of 6 months of data acquisition for local access.

The AdV CCs receive a copy of the data and provide storage resources for permanent data archiving. They must guarantee fast data access and computing resources for off-line analyses. Finally, they must provide the network links to the other AdV computing resources.

For this goal a robust data distribution and access framework (based on file and metadata catalogs) is a crucial point.

The AdV collaboration manages also smaller CCs used to run part of some analyses, simulations or for software developments and tests, in which the CM does not foresee specific data transfer and access frameworks.

During science runs the Cascina facility is dedicated to data production and to different detector characterization and commissioning analysis, which have the need to run "on-line" (with a very short latency, from seconds to minutes, to give rapid information on the quality of the data) or "in-time" (with a higher latency, even hours, but which again produce information on the quality of the data within a well defined time scale). The detector characterization activity gives support to both commissioning and science analysis.

Science analyses are carried out offline at the AdV CCs, with the only exception of the low-latency searches, as explained before. Some analysis, due to the fact that we analyze data jointly with aLIGO for many searches, are carried on in aLIGO CCs.

To face the huge computational demands of g.w. searches in ADE, there will be the need to gather the resources of many CCs into a homogeneous distributed environment (like Grids and/or

Clouds) and to adapt the science pipelines to run under such distributed environment.

Another very important need is to provide a Grid-enabled, aLIGO-compatible Condor cluster for AdV people.

Another important task, which we started to face, is the possibility to run search pipelines in GPU clusters.

The AdV CM will continue to guarantee (in at least one CCs where the raw data are archived) local access to the data and to the computing resources, as requested for the offline detector characterization studies and for software development and testing purposes.

Most g.w. searches require the use of a network of g.w. detectors (at least AdV and aLIGO). As a consequence, these search pipelines must be able to run either in AdV or aLIGO CCs. It is therefore important to develop pipelines adaptable to different environments or interfaces which hide the different technologies to the users.

The most important issues addressed by this model may be summarized as follows:

- guarantee adequate storage and computing resources at Cascina, for commissioning, detector characterization and low-latency searches;
- guarantee fast communications between Virgo applications at Cascina and aLIGO CCs/other detectors for low-latency searches;
- guarantee reliable storage and computing resources for off-line analyses in the AdV CCs;
- push towards the use of geographically distributed resources (Grid/Cloud), whenever appropriate;
- push towards a homogeneus model for data distribution, bookkeping and access.

Figure 1 gives a big picture of the data workflow for what concerns scientific data analysis (DA) and detector characterization (Detchar) activities in AdV.

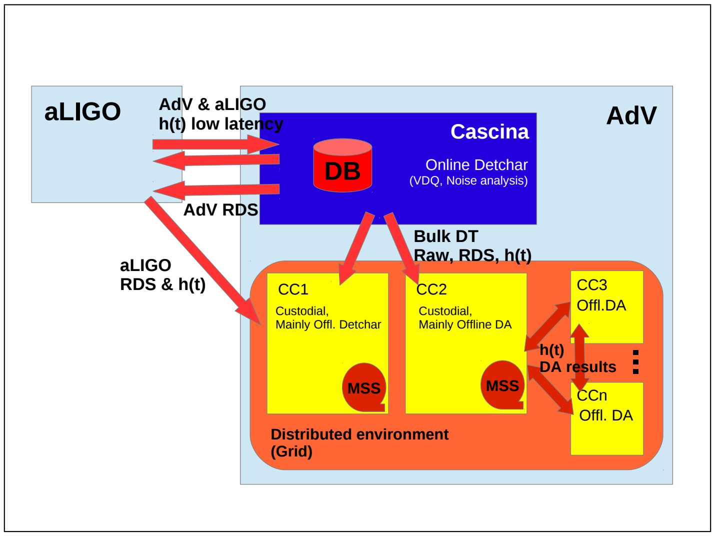

Figure 1: Data workflow for DA and Detchar activities in AdV.

Part I

Advanced Virgo Computing Model

Chapter 1

Computing and Data Analysis workflows

1.1 Introduction

The Advanced Virgo (AdV) data analysis activities can be classified into three main categories.

- Commissioning;
- Detector characterization (calibration, data quality, noise studies);
- Scientific analysis (Burst, CBC, CW, Stochastic).

And the work in each category follows a different workflow. Beside this, different workflows result from the “on-line” (and “in-time”) or “off-line” application of the analysis.

In this document only the data related to the above activities are mentioned.

Different kinds of data are produced by the detector at the EGO site in Cascina (“Tier-0”). All the commissioning and detector characterization activities are performed in Cascina, on different data sets and with different latencies. The workflow for these is described in Sections 1.1, 1.2 and 1.3.

The Cascina facility is dedicated to data production (during the runs) and to commissioning and detector characterization analysis, which have the need to run “on-line”, with a very short latency, from seconds to minutes, to give rapid information on the quality of the data, or “in-time”, with a higher latency, even hours, but which produce information on the quality of the data within a well defined time scale. The detector characterization analysis give support to both commissioning and science analysis. The only scientific analysis performed in Cascina are the “low-latency” searches, which aim to provide fast alerts to the astronomical community in order to perform follow-up analysis of candidate GW signals. All the other scientific analysis are carried out off-line and not in Cascina.

The workflow of the Scientific analysis is described in Section 1.4.

Let’s clarify that in what follows all the data to which we refer are data taken during runs labelled as “Commissioning (or Engineering) runs”, “Scientific runs” or “Astrowatch runs” . “Astrowatch runs” are those runs when, even if the sensitivity or the data quality of the detector will not be such to have a Scientific run in place, the joing aLIGO and AdV collaborations will decide to use the data for some scientific purposes.

In this document we will indicate with “AdV CCs”, or simply CCs, the external CCs, where the offline analyses are done, of the collaboration (that is, not including the Cascina/EGO farm). We will then explicitely mention the Cascina farm or aLIGO CCs, where needed.

1.2 Commissioning and operation workflows

The data whose input is DAQ are referred here as the "primary data producer".

Their workflow is at the basis of all the commissioning, calibration and scientific data production, detector characterization and scientific analysis and is shown in Fig. 1.1.

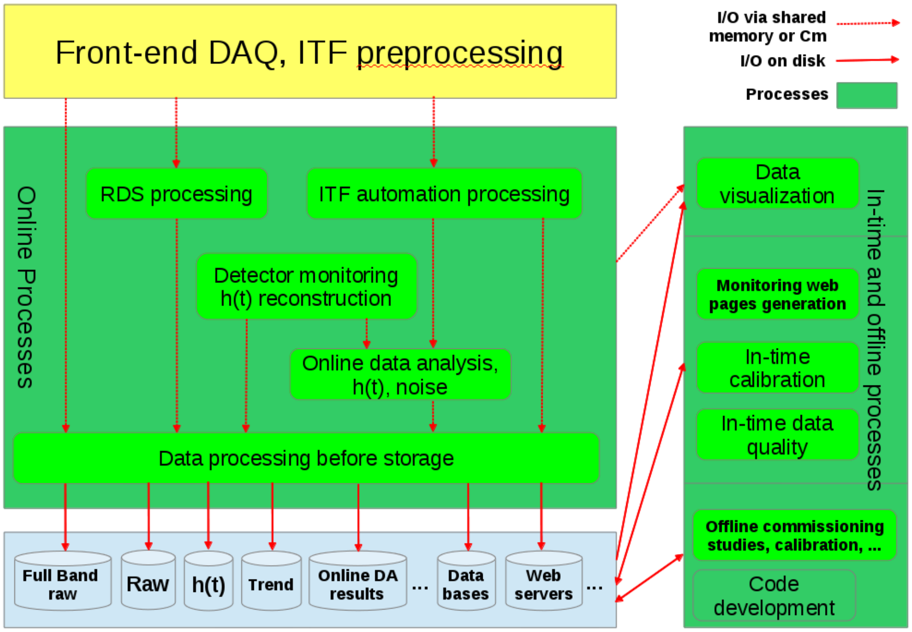

Figure 1.1: Data workflows in Cascina for commissioning, detector characterization and "on line DA results" (low-latency results)

The raw data are collected, formatted and merged in the real-time processes of the front-end data acquisition. Part of the data can be provided to the automation system to control the ITF or reduced to build the different data streams.

These data are written in very short frame files, to be accessed online by experts and commissioners, with low latency (less than 10 s). Short frame files are not stored on disk. They are used for detector monitoring, online reconstruction of the GW signal and online data processing (data quality, low-latency data analysis). The data are available online for user access within less than 30 s.

The raw data and data produced by the different online processes are finally put in different streams for storage. The final files are readable on disk with latencies from 2 minutes to 30 minutes, depending on the streams. They are then available for offline use. The online processes building the different non-reproducible data streams are critical and special care should be taken to prevent any impact from other activities. The in-time and offline commissioning processes and developments access the main data streams on disk.

The raw data is combined with auxiliary measurements and models to build a time series representing the gravitational-wave strain signal ("h(t)").

This is then calibrated, and also flagged for quality control veto and cleaning. In addition, there are a large number of auxiliary instrumental and environmental monitoring channels that are also ingested.

The commissioning activity include both on-line/in-time computing during data taking, and offline studies and development after data taking. The general workflow is shown in fig. 1.1.

In order to monitor the interferometer, different data streams are built in the data collection system architecture. They can be accessed on-line for DAQ debugging, ITF or sub-systems commissioning, or any other needs, with different latencies.

- On-line workflow for commissioning and detector characterization

  The "online short frames" (< 10 s) are used to reduce the access latency to a few seconds.

- In-time and offline workflow for commissioning and detector characterization

  After compression, the main streams are finally stored on disk for offline use. For each stream, the size of the frames and the number of frames per file are adjusted before storage in order to speed up the data access from files. In general, the frame length of data on disk is increased. The in-time and offline commissioning processes and developments access the main data streams on disk.

From the Virgo experience, the commissioning tasks are not deterministic: they include development, interactive use, manual processing, simulations, access to data, web servers and databases. This induce variations in the usage of the available machines and of the data access load. Lots of processes, both in-time and offline, need large I/O access to data stored on disk: the data access should not be a bottleneck.

Commissioners from outside the EGO site need fast remote accessibility, including graphical tools for data visualization.

1.3 Detector characterization workflow

Most of the Detector Characterization analysis and detector status monitoring must be done on-site (Cascina) and with a latency which can vary from few seconds (on-line analysis, for transient signals analysis) to less than one day (in-time, for noise line identification, noise correlation, non linear analysis).

These basic analysis must be helpful for commissioning activity but also for astrophysical searches, since their results are used in the data cleaning procedures. The basic scheme is again given in fig. 1.1, where also the commissioning and "low-latency" workflows have been reported.

The detector characterization workflow is divided into two main areas:

- Data Quality
- Noise studies

1.3.1 Detector Characterization: Data Quality

The Data Quality work includes glitch studies, online vetoes production, offline vetoes production and the development of tools for monitoring, investigations and commissioning help. The main axes of this work are:

- An online trigger generator and an online veto production, which will be run on a set of computing nodes, having in input online frame data from DAQ. The output of those processes is stored in frames and/or in a specific format (ROOT files for the Omicron triggers, DQSEGDB database for the online vetoes).

- Several off-line or in-time tools, run periodically or on user's demand, for commissioning and glitch investigations. Those tools need to access raw data, trend data, RDS data, spectro data or the DQSEGDB database. When those tools run automatically and periodically, they will produce results (plots and web pages) daily archived in the web area.

- A database to store the Data Quality (DQ) flags from LIGO and Virgo: the "LIGO-Virgo Data Quality Segments Database" (DQSEGDB). This is a MySQL DB. We will have two instances of DQSEGDB (one at CIT, one at Cascina) always containing the same information, so that queries done to one or to the other will be equivalent. DQSEGDB will contain also Science flag, Lock flag, Injection flags. The DQSEGDB server answers to queries by sending the result in a standard format and the client receiving this result will be able to convert it in a user-readable format.

- A set of scripts to easily manage the reprocessing of the data quality flags and the reprocessing of the Omicron triggers. This will be done at the Lyon computing center and will include the maintenance of the needed software packages and the management of the needed storage at the Lyon CC.

1.3.1.1 Omicron pipeline

The pipeline runs online over about 600 channels from the raw data stream. It produces Trigger files in ROOT format, to be used for various features of data quality and glitch investigation (DQ flags performances, glitch rate monitoring, Omiscans...). Plugins like UPV will be added to Omicron pipeline and will produce useful data quality information and, as much as possible, online vetoes to be stored in the DQSEGDB.

1.3.1.2 On-line vetoes

Those are Data Quality (DQ) segments that will be produced either by online processes (for instance UPV and Excavator) that will use Omicron triggers, or by specific processes like BRMSMon or by the processes used in the Detector Monitoring System (DMS). All those online vetoes will be propagated to online analysis and stored in DQSEGDB. They will be the official DQ segment lists used by offline analysis until a set of DQ segment lists is reprocessed offline and stored in DQSEGDB.

1.3.1.3 Detector Monitoring System (DMS)

This is a set of processes taking as input data the DAQ raw data stream. Those processes produce DQ flags used to provide in control room a complete online monitoring and alarm for the various interferometer's subsystems and for the processes running in the DAQ and in the various online processing tasks. Those DQ flags can also be used as online vetoes and thus stored in DQSEGDB.

1.3.1.4 Spectrograms

A set of spectrograms over one day or one week is created (SpectroMoni pipeline), periodically updated and available on web pages within the MonitoringWeb area. The inputs are selected raw data channels, from the DAQ raw data stream. The spectra are computed on-line and saved under frame format in a specific "spectro" data stream, stored on a dedicated disk area. The various plots created from those spectro data are computed hourly and archived daily and represent most of the CPU usage and a significant part of the disk usage of the MonitoringWeb framework.

1.3.1.5 MonitoringWeb

This is a general framework which handles monitoring information and plots produced by various scripts, mostly using the trend data, spectro data, Omicron triggers or DQSEGDB entries. They give information on the interferometer status and on all the ongoing on-line data quality and science analysis. Information given are, e.g., the General Status of the Interferometer, Locking, Vacuum, Infrastructure Monitoring, DAQ, Noise Budget, Spectrograms, MBTA triggers, Omicron triggers, Online DQ, etc...

1.3.1.6 DQ developments

Some work is needed to test new developments, using off-line raw data. The output of these studies are investigations and pipeline improvements, so no data distribution is foreseen. Such work needs anyway some disk space to store temporary output data and a good access to the raw data and Omicron triggers.

1.3.2 Detector Characterization: Noise studies

The noise studies work is focused on a general description of noise features. The individual noise monitoring tools are generally referred to as Noise Monitors (NM). The NM are plugged-in to a general coherent framework, the Noise Monitor Application Programming Interface (NMAPI), which enables results produced by each NM to be queried and presented via a web browser, as shown in Fig. 1.2.

NM can be grouped in the following way (see 1.1):

- On-line NM tools - consumers of data from the online DAQ chain;
- In-time or off-line NM tools - consumers of data written to disk.

Typically, these algorithms or pipelines produce output data which can directly be plots or web pages. Results can be also stored in ASCII or binary files, or archived into a MySQL database.

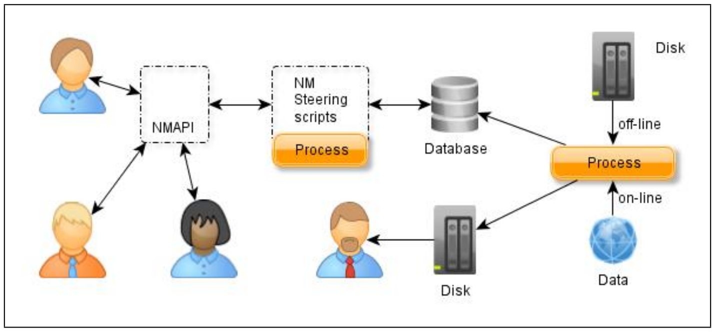

Figure 1.2: NMAPI framework

The framework is based on the idea of distributing the computational work to different computing nodes, using batch system facilities, allowing the users to access either to summary pages or to scripts for specific queries. In Fig. 1.2 the 'process' represents any NM pipeline described hereafter.

1.3.2.1 NoEMi

NoEMi is a tool for the in-time discovery and follow-up of frequency noise lines and narrow band disturbances in the ADE data. It analyzes raw frame files (the h(t) channel, the raw Dark Fringe (DF) channel and a subset of environmental monitoring sensors) looking for matching frequencies and similar patterns between the lines found in the science data and the environmental sensor data.

It runs every night on the data collected in the previous day. It generates daily web pages reporting on the run data quality and it feeds the Lines database, which is used in the vetoing procedures of the CW and Stochastic searches.

NoEMi will also produce an online version of the input data files for the CW Hough analysis: the SFDB files and the Peakmap files, as explained in Section 1.4.

It will run at Cascina, on the local Condor batch system.

1.3.2.2 SILeNTe: System Identification Linear et Nonlinear Techniques

SILeNTe is a non-linear system identification technique developed to identify linear and non-linear noise coupling mechanisms. It is an in-time analysis. The main feature of the method is that it can compute the specific contribution to the model of a channel, or a combination of channels, through the process of orthogonalization with respect to any other channels contribution. The application is intended for on-demand noise analysis. The input are raw files, the results are given as a set of plots and text files which can eventually be stored. The plots will be inserted in NMAPI.

1.3.2.3 Regression monitoring

It is an in-time analysis. This monitor allow to survey the bilinear coupling between different auxiliary channels and the data channel. The input list of noise lines to be analyzed by the pipeline will be generated daily by NoEMi. The auxiliary channels to be analyzed will be selected among those expected to produce bilinear coupling, given the available computational resources. The input are raw frame files. The analysis runs on the local Condor batch system. The plots will be inserted in NMAPI framework.

1.3.2.4 WDF

WDF finds triggers associated to transient signal events. It analyzes data in the time domain, using a wavelet transform, to find an excess of power in the data and identify the trigger.

The transient signal events are produced on-line and all the parameters which characterize the event are stored on line in a MySQL database in Cascina.

This pipeline is meant to give in real time information on glitches rate. Since it inserts triggers directly in a MySQL database in on-line mode, it could be useful to test a framework for a glitches database.

The inputs are selected raw data channels, from the DAQ shared memory or from disk, the output are MySQL entries, plots and html summary pages.

1.3.2.5 Coherence

This analysis in in-time analysis. Coherence reveals the coherence between the dark fringe and auxiliary channels. The input are raw frame files, the output MySQL entries and plots.

1.3.2.6 Non stationary monitoring

The NonStatMoni pipeline run on-line to monitor band-limited RMS in many bands BRMS and showing slow variations. The inputs are selected raw data channels, from the DAQ shared memory, the output are html summary pages and plots.

1.4 Science data analysis workflows

Here we describe the workflow of the scientific analysis. It is important to clarify that the searches (for scientific motivations) with the exclusion of the CW, are done by jointly analyzing the data of all the detectors of the network. Thus, in these cases, the offline analysis are done in aLIGO or in AdV CCs.

The resource sharing and the division of the tasks will be jointly agreed by the LVC DA groups, as required by the LV MOU (draft, still to be approved, are LIGO-M060322-v3 and VIR-0324A-11 [4]). It is important to understand the constraints of each Collaboration's computing resources, and to optimally adapt the available resources to provide the appropriate degree of interoperability of computers and software and to aim at the broadest use of hardware and software.

AdV DA groups should in any case develop or contribute to pipelines which do not show architecture constraints such to preclude to carry on the analysis in AdV CCs or in the Cascina farm (on-line and in-time pipelines).

1.4.1 Low latency searches

1.4.1.1 Burst

Science goal: a prompt identification and reconstruction of transient (un-modeled waveform, duration up to a few seconds) GW signal candidates, including estimates of related false alarm rate, source localization, waveform reconstruction, and detection efficiencies for some pre-determined signal classes.

These informations are available on a timescale of the order of one-few minutes and are used to trigger a procedure to alert external telescope partners.

Name of the pipeline: 2^{nd} Generation online *coherent WaveBurst (cWB)*. We will call it simply "cWB".

Input data are the online frames, h(t) and status flags.

Output data are triggers to GraceDB, summary web pages and ROOT or text files.

A brief description of the workflow follows:

- input data are detector' s h(t) online frames and online DQ flag vectors. Plus a dedicated Mock Data channel with simulated signals.
- separate analysis run per each detector network configuration and for different signal polarization states, to evaluate the distribution of accidentals, detection efficiency and uncertainties in signal reconstruction.

cWB online is planned to run on LIGO clusters, at CIT (California Institute of Technology). At present there is no plan to run it at Virgo (Cascina).

A second pipeline, for an in-time search for GW burst candidates triggered by external astrophysical events, in particular by Gamma Ray Bursts (GRBs), will be in place in ADE, but the workflow is not yet ready. Thus it will be added in a next version of the CM.

1.4.1.2 CBC

Science goal: Low-latency detection of compact binary coalescence signals.

Low-latency detection and sky localization of coalescing binaries, especially those involving at least one neutron star, will allow us to quickly pass on triggers to electromagnetic partners to look for possible electromagnetic counterparts.

Name of the pipeline: MBTA ("Multi Band Template Analysis")

Multi-Band Template Analysis (MBTA) is mostly mean to be a online pipeline, although it can also be run in offline mode. It can run on a single detector with the goal of detector characterization and data quality studies or on multiple detectors to provide triggers for EM follow-up.

The MBTA pipeline uses the AdV DAQ tools to access the data (the FdIO library) and therefore usually runs online at Cascina. It takes as input the online frames, h(t) with status flags. Triggers are provided as frame files and interesting triggers are submitted to the GraceDB database.

Location Environment Architecture: In ADE, MBTA will run online in Cascina. It will also run offline reading frames on any computing center from the command line for test purpose or with a batch system.

Output data are trigger frame files and entries in the GraceDB database.

1.4.2 Off-line searches

1.4.2.1 Burst

Burst signal searches are also performed offline to set the best achievable astrophysical results, taking advantage of the improved knowledge available offline on the Data Quality, calibration, False Alarm Background noise and detection efficiency studies. The top science goals for offline burst searches are:

- All-sky all-times offline search using *coherent WaveBurst* pipeline;
- Gamma Ray Burst triggered search using *X-pipeline.*

1.4.2.1.1 All-sky all-times offline search using "coherent WaveBurst" pipeline .

Science goal: identification and reconstruction of transient (un-modeled waveform, duration up to about 10 seconds) GW signal candidates, including estimates of related false alarm rate, source localization, waveform reconstruction, and detection efficiencies for selected signal classes.

Name of the pipeline: 2^{nd} Generation offline *coherent WaveBurst (cWB).*

The goal of the analysis implementation is to produce possible GW candidates within about two months from the related data taking, more specifically within one month after final DQ and calibration information is made available.

Standard input data are the h(t) frame files of all the detectors of the network and offline DQ segments, obtained with a query to DQSEGDB. Output data are candidate triggers, whose parameters are written in ROOT or text files. The workflow of the analysis can be customized using user defined plug-ins and/or different tools. Separate independent analysis will be run per each detector network configuration and for different signal polarization states.

The main standard analysis procedures are two:

1. search for signals and for the distribution of accidental false alarms,
2. search for fake simulated signals summed to the h(t) data (to estimate the detection efficiency of the search and the uncertainties in signal reconstruction). In this procedure additional inputs comes from Mock Data Challenge (MDC) frame files of software signal injections or tables of selected software signal injections. MDC frame files are either produced by cWB itself or by burstMDC, which is a LIGO dedicated pipeline running at aLIGO CCs.

The cWB offline pipeline can make use of an optional pre-conditioning module of input data (cWB pre-conditioning). This module inputs h(t) frame files, raw data frame files and DQ segments. The outputs are de-noised h(t) frame files and ROOT files summarizing the de-noising performances. The de-noising is performed separately offline for each detector, so that for AdVirgo it will run at CC2 producing de-noised AdV h(t). The de-noised AdV h(t) frame files can then be used as alternative input of cWB as well as other pipelines, instead of the original AdV h(t) frame files.

The schematic work flow is shown in Fig 1.3.

1.4.2.1.2 All-sky all-times offline search using *STAMP all-sky* pipeline .

Science goal: extend the all-sky all-time offline search performed by cWB to un-modeled waveform transients of duration of the order of hundreds of seconds.

The target timeline of production of results is matched to the within the same

Name of the pipeline: *STAMP all-sky.*

The pipeline is at an early stage of development and its structure is not yet final. Input data are h(t) frame files of all the detectors of the network and offline DQ segments, obtained by queries to DQSEGDB. Output data are candidate and false alarm triggers. Predictions of data managment needs and computational loads are still very uncertain.

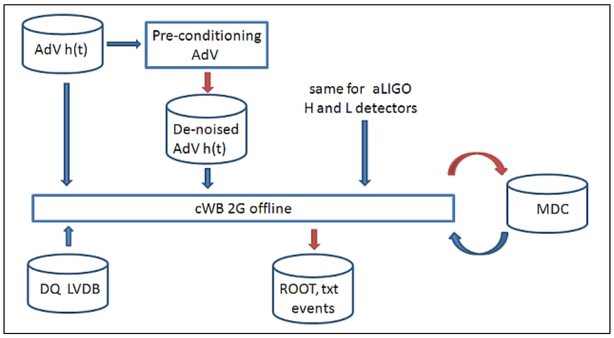

Figure 1.3: Workflow for the cWB offline pipeline. Separate analysis will be run per each detector network configuration and for different signal polarization states. The cWB pre-conditioning are optional modules

1.4.2.1.3 Gamma Ray Burst triggered search using *X-pipeline* .

Science goal: identification of transient GW signal candidates related to GRB events detected by X or Gamma ray satellites. The pipeline searches for any GW waveform with circular polarization and signal duration up to about a few seconds incoming from the direction of the GRB. The goal is to produce results within one day from the well characterized GRBs, and then confirm/complete the results on all available GRB triggers offline n a similar timeline to the all-sky search.

Name of the pipeline: *X - pipeline*.

X-pipeline will run in two modes per each GRB trigger:

1. on the on-source time window including GRB time to identify GW candidates
2. on the off-source time window not including the on-source but close to the GRB time to estimates the false alarm rate and detection efficiencies for selected signal classes. This latter mode dominates the computational load of the search.

The code is compiled under MATLAB. Input data are the h(t) frame files of all the detectors of the network and offline DQ segments, obtained by queries to DQSEGDB. Output data are candidate and false alarm triggers, whose parameters are written in MATLAB files and web pages. The software signal injections to test detection efficiency are produced on-the-fly internally by the pipeline.

1.4.2.2 CBC

The offline CBC analysis workflow is divided into the following areas:

- Detection of compact binary coalescence signals
- Extracting parameters, testing GR, and determining the neutron star equation of state with compact binary coalescence detections

1.4.2.2.1 Detection of compact binary coalescence signals .

Science Goal: Detection of signals from coalescing compact binaries, two neutron stars, a neutron star and a black hole, or two black holes.

Pipeline: ihope pipeline, augmented with GWTools.

ihope is mostly an offline pipeline, although it has been used in semi-online mode ("daily ihope"), partially as a diagnostic tool. Input data are the h(t) frames of all the detectors of the network and the DQ segments, obtained with a query to DQSEGDB.

ihope is a workflow with executables plugged in for a list of tasks, mainly: template constructions, matched filtering, background estimation, trigger production.

Executables can be combined and/or replaced by more efficient ones. An example is *GWTools*, an OpenCL-based algorithms library with both CPU and GPU capability for (among other things) template bank generation, matched-filtering, and χ^2 calculation.

For 1 year 's worth of data, ihope writes a total of $\sim$ 4TB to disk, comprising a large number of intermediate data files and an HTML summary.

Location Environment Architecture: ihope and GWTools will run at Bologna and Lyon using CREAMCE/Pegasus for job submission.

Output data, referred in Sect. 2.4 as "ihope and GWTools output data" are xml files for template banks, triggers, and injections. And HTML summary pages.

1.4.2.2.2 Extracting parameters, testing GR, and determining the neutron star equation of state with compact binary coalescence detections .

After each compact binary coalescence detection, we will want to estimate the parameters of the source and

(a) Parameter estimation (LALInference pipeline)

(b) Test the strong-field dynamics of gravity (TIGER pipeline).

(c) assuming GR is correct, determine the equation of state of neutron stars with coalescences that involve at least one neutron star (TIGER pipeline).

LALInference and its extension for testing GR (TIGER) are in principle offline pipelines, but LALInference can be called by an online pipeline for rapid sky localization. The input data, when used online, is obtained from the so-called GraceDB database of online triggers, and analysis results can be pushed to GraceDB.

h(t) frames and DQ segments are also needed as input to this analysis.

LALInference jobs produce samples from the posterior distributions of the sample PDFs, stored as ASCII text files. To store the results of a 1 year run, typically 0.5 TB are needed.

Parallel jobs can produce multiple instances of these to increase accuracy, producing up to 500 MB of intermediate data.

These are digested into web pages stored on the clusters web server in the users public html directory, each of which is around 30MB in size at present.

In the TIGER configuration, a large number of injected waveforms ($\mathcal{O}(10^6)$) also need to be analyzed to determine the "background" distribution of log odds ratio for pure GR signals.

The Location Environment Architecture is presently LDG ("LIGO data GRID") Condor cluster, but the pipeline for AdV will run at Lyon and Bologna using Pegasus for workflow submission.

Output data are Posterior samples, posted as summary HTML pages.

1.4.2.3 CW

The workflow for the CW searches is divided into four different main areas, which reflect the Science goal beyond it:

- All-Sky searches for unknown isolated neutron stars.
- Targeted searches for known isolated neutron stars
  (frequency, spin-down and position are known).
- Direct searches, for isolated neutron stars at known position
  (only the position is known).
- Searches for binary neutron stars (All-Sky, directed, targeted).

The nature of this search is such that it can be carried on using only data from one detector. Thus, the priority is given to the analysis of AdV data. Obviously, analyzing data from more detectors allows to improve the search sensitivity, then including in the analysis also data from other detectors will be the next step. The noise artifacts to be removed are not the same which give problems to the other searches. Thus the procedure to assess the quality of the data is embedded in the analysis itself, and done by reading the “status flag” channel embedded in the h(t) frame files. Only the outcomes of the NoEMi pipeline (list of known or unknown spectral lines in the detector) are used for this search.

The analysis is typically run off-line when several month of data is available.

1.4.2.3.1 The All-Sky search of unknown neutron stars .

Goal: Search for unknown rotating neutron stars.

I) Frequency Hough search (Periodic Source Search, PSS)

Input data are the g.w. h(t) frame files (with “status flag” channel inside). And “Ephemerides” data, obtained from JPL and elaborations in PSS. The first outcome are 4 sets of files which contain the FFTs (“Fast Fourier Transform”) data base, in different frequency sub-bands, of time duration which depends on the maximum frequency of the band. These are binary files: “SFDB (PSS search)” data. From these we produce time/frequency “Peakmaps (PSS All-Sky)” data. The peakmaps are the input to the main search code, the Frequency Hough transform pipeline which produces the “Candidate (PSS All-Sky)”. Candidates from different runs of the detector or from different sub-periods of one run, or even from different detectors are the input to the Coincidence procedure which again produces “Candidate files”. On these we run the Follow-up procedure, which uses “Follow-up peakmaps (PSS All-Sky)” and produces “Follow-up results (PSS All-Sky)”. The best architecture where to run this search is GRID. But the codes might also run under native batch systems.

The format of these data (SFDB, peakmaps,candidates) is defined in the PSS libraries, where the basic functions to write and read them are given. Fig. 1.4,left, shows the workflow for this search.

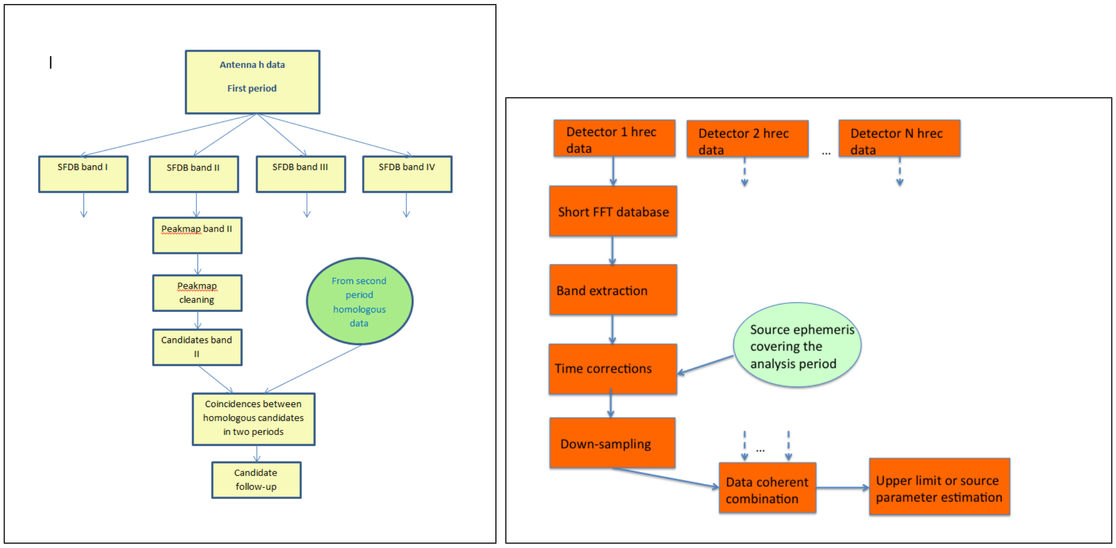

Figure 1.4: Workflows Left: for the PSS CW All-Sky search. Right: for the PSS targeted search

II) Polgraw AllSky pipeline

Input data for this search are the “SFDB (PSS All-Sky)” data. Using PSS software 2-days / 1Hz segments of sub-sampled data are produced. These 2-day time domain sequences are input data for the Polgraw AllSky pipeline. There are no architecture constraints to run the code. The analysis consists of two steps. The first step analyzes the “2-day segments (Polgraw All-Sky)” and produces candidates, “Candidates (Polgraw All-Sky)”. The second step consists of searches for coincidences among candidates obtained in the first step over the course of a data run with consistent source parameters. There is also an additional cleaning and candidate selection. The final result are files with coincident candidates.

1.4.2.3.2 Targeted searches for known neutron stars .

Goal: search for known pulsars, identified by precise values of position, frequency and frequency derivatives (and possibly also intrinsic velocity respect to the line of sight) .

Ephemerides of the known pulsars files from the electromagnetic observations are needed to run this search.

I) "Rome Targeted" PSS pipeline

The input for this pipeline are the "SFDB (PSS search)" data. Given a target pulsar, the analysis consists of several steps. First, from the SFDB a small band around the signal expected frequency is extracted, producing an "SBL (PSS Targeted)" (single block data format) file, with also other relevant information. this is the input to the main pipeline, which finally produces "Corrected time series (PSS Targeted)" data, form which the results of the analysis are obtained, stored in one output file (of negligible size) with the upper limit for the non-detection case or with the signal parameters estimation.

The analysis method can easily handle data from multiple detectors that can be coherently combined in order to increase the search sensitivity, in which case the procedure is repeated over the different data sets.

The pipeline is written in Matlab and the analysis can be run on whatever machine where Matlab is installed, or alternatively it can run using a compiled version of the software.

The workflow for this pipeline is given in Fig. 1.4 (right).

II) Polgraw Targeted Search pipeline

The input data for this search are coarse or fine heterodyne time domain data produced by Glasgow-LSC pipeline and then copied to our clusters. Also the codes can be adapted to use narrowband time domain sequences produced by Rome PSS pipeline. Once the heterodynes are available they are downloaded to local clusters of the Polgraw-Virgo group and analyzed with Polgraw Targeted Search pipeline. If the computed value of the produced statistics is not statistically significant the output result is a file, "Search results (Polgraw coherent)" with the upper limit, obtained by injecting signals to the data with random parameters. If the signal has been detected a file with the estimated parameters is produced, again "Search results (Polgraw coherent)". In both cases, the sizes of these files are very small.

The pipeline does not have any architecture constraints, and it might run under Condor LSC clusters or under any native batch system at Bologna or Lyon.

1.4.2.3.3 Direct searches, for isolated neutron stars at known position .

Goal: search for a periodic source with a known position (or with a small position uncertainty) in the sky.

I) Pisa pipeline

The search is performed over a small bandwidth around a reference frequency.

Input data are the g.w. $h(t)$ frame files (with "status flag" channel inside), tables containing the list of time segments to be analyzed and ephemeris data.

The main output data are the final spectra, over a bandwidth of a few Hz.

The amount of data produced is of the order of 450 GB to process 1 yr of data (having considered the case of a search over a larger bandwidth).

Fig. 1.5 shows the workflow for this search.

The pipeline is implemented in C++, starting from prototypes tested in the Matlab environment. From the C++ code python bindings are obtained, and the jobs consist physically in python scripts that can be easily configured and modified. The pipeline is designed to be used with a standard job scheduler. It will run under GRID.

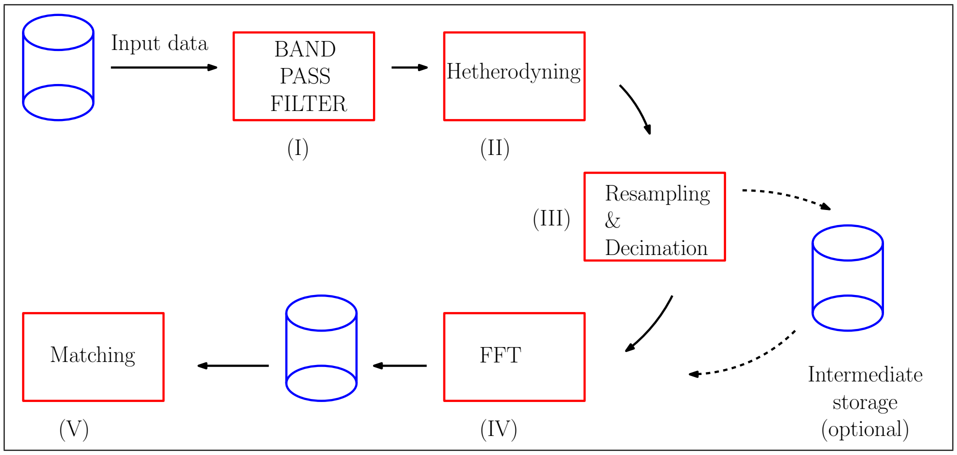

Figure 1.5: The workflow implemented in the pipeline. The initial data are (I) filtered with a narrow band pass filter, (II) hetherodyned to move the center of the filter band to dc, (III) resampled to compensate doppler and spin-down, (IV) transformed to the frequency domain and (V) post-processed.

1.4.2.3.4 Searches for binary neutron stars A pipeline was developed by Virgo group at Nikhef for an All-Sky search for binary neutron stars. It is called "Polynomial Search", as in teh method the phase of the signal is approximated as a third-order polynomial in time within a single short-time Fourier transform (SFT). There are currently plans to develop an additional analysis stage that combines results for successive SFTs.

In the first stage of the analysis, the input data is Fourier-transformed in segments of fixed time length. Then the average power spectral density is computed. Finally, a large number of polynomial signal templates is matched against the input data in the frequency domain, yielding the correlation as a function of phase for each set of polynomial coefficients.

1.4.2.4 Stochastic

Goals: The primary objective of the stochastic group is to use cross-correlation techniques to measure (or set an upper limit) on the stochastic gravitational-wave background (SGWB), assumed to be created by the superposition of a large number of weak unresolved sources, and expected to be below the instrumental noise of a single detector. It is important to notice that unlike for other types of sources, the Virgo data cannot be analyzed alone. The sources can be of either cosmological (inflation, phase transitions, cosmic strings...) or astrophysical origin (compact binary coalescences, supernovae, core-collapse to black holes, rotation neutron stars).

The second objective is to extract informations from these measurements doing **parameter estimation**.

Assuming the SGWB is unpolarized, stationary and Gaussian, and that the noise in the two detectors is Gaussian and independent from each other and from the GW signal, three types of analysis are performed :

- the all sky/isotropic search (the standard cross-correlation algorithm)
- the radiometer search (directed search)
- the all sky/non isotropic search (based on spherical harmonic decomposition)

The stochastic group is also using the infrastructure of stochastic searches to search for periodic searches such as Sco X-1 (the radiometer search) or long-lived unmodelled gravitational-wave transients with the Stochastic Transient Analysis Multi-detector Pipeline (STAMP) (see burst section)

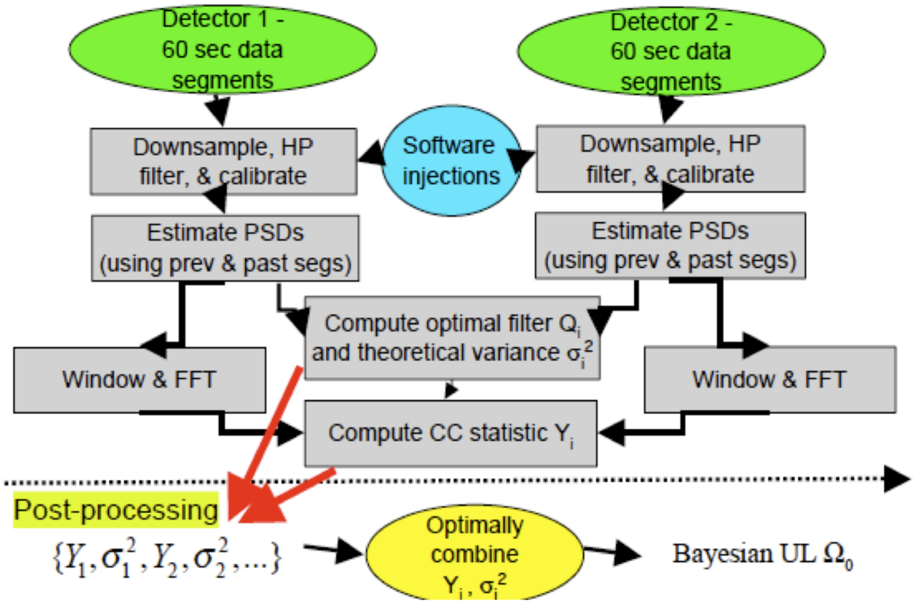

Figure 1.6: Workflow for the Stochastic Isotropic analysis

Finally, there is theoretical evidence, that non stochastic background signals from astrophysical origin or cosmic strings may be non-Gaussian, and the stochastic group is currently developing Non Gaussian algorithms.

In the following, we have divided the workflows into two main schemes:

- Isotropic analysis (based on standard cross-correlation methods)
- Spherical harmonic analysis, with the special case of Radiometer (or targeted) search.

In all the above cases, the input data to the analysis are the g.w. h(t) frames of all the detector in the network, with "status flags".

The codes for these searches are written in MATLAB, compiled with the Matlab compiler to produce a C executable. The main workflow is almost the same for all the analysis. The output files are text or mat files which contains the cross correlation product for the time segments and the theoretical error.

1.4.2.4.1 The isotropic analysis Fig. 1.6 shows the workflow for the analysis.

Input data are divided into 60 s segments and for each segment: read g.w. h(t) for IFO1 and IFO2, down-sample, high pass filter, apply frequency mask. Then: calculate the strain noise power spectral densities, calculate optimal filter, calculate cross correlation estimator and theoretical variance. Finally, derive the point estimate and its standard deviation. These quantities are used as parameters for the posterior probability distribution from which the final upper limit is computed.

1.4.2.4.2 The spherical harmonic analysis .

The steps up to the frequency mask application are the same of the isotropic search. Then we calculate the cross and auto power spectra (C and P) in a spherical harmonic basis (the dirty map) , invert Fisher matrix, calculate the GW power estimator in a spherical harmonic basis (clean map). For the isotropic search, the workflow is the same but we look at a specific direction is the sky and use an overlap reduction function which depends on the direction rather than a sky average.

Part II

AdV Data Model

Chapter 2

Data Model: from production to processing

2.1 Introduction

This part of the AdV Computing Model focuses on a “Static Vision” of the data sets produced by the detector (ITF)

2.2 Data sets in IGWD Frame Format

We give here the description of the ITF data set which is produced online during a run of the ITF, in Cascina. These data serve as input for many different studies and thus for many analysis pipelines. The “IGWD” format (“Interferometric g.w. detectors” format) used to represent the data. The data is a stored as a collection of “frames”. A frame stores the time series data of various channels having different sampling frequencies. It is described in Sect. 5.5. The necessary storage has been estimated using a reference time of 1 yr, with 100% duty cycle. With “Local” here we mean data which are in Cascina, with “Exported” data copied to one or both the CCs, and with “CC” data created at the CCs.

2.2.1 Online frames. Not stored

In order to reach very short latency for online processes like commissioning, detector characterization and low-latency searches very short “online frames” are created. They are stored in shared memories with buffer depth of few seconds to minutes. These frames include h(t) , quality flag and auxiliary channels.

2.2.2 Full bandwidth raw data stream. Local

A full bandwidth raw data stream will be built from the front-end DAQ and automation processes and directly stored on disk without any data selection nor decimation. It is also used as a debugging dataset. The current estimation of the AdV flow is 4 TB/day. A depth of 3 days of the full raw data stored at Cascina is needed for debugging of the digital DAQ front-end and automation. It represents a buffer of about 12 TB. These data are not transferred to the CCs.

2.2.3 Raw data stream. Local. Exported

The raw data stream is built from all the acquired channels, with some decimation. Beside the main gravitational channel, the so-called Dark Fringe signal, and the equivalent calibrated channel (h(t)), a large number of auxiliary signals which are used to lock and control the interferometer are acquired. To keep under control the environmental or intrinsic noises ≈ 100 sensors are distributed around the main building and the detector

(see the information at : https://wwwcascina.virgo.infn.it/EnvMon/sensors.htm).

All these sensors produce signals which are part of the raw data stream. The channel with the "status flag" of the detector is also stored here. The h(t) stream computed online is merged into the raw data stream, to ease the work of of all those analysis which need to jointly analyze raw data and the g.w. data (it represents 0.4% fo the raw data).

In addition to these standard channels, 10 s of not-decimated data will be stored every 1000 s in the raw data stream (these data correspond to a subset (∼ 1%) of the *full bandwith raw data*, included in the raw data for long-term monitoring and storage).

The current estimation of the AdV raw data stream flow is 2 TB/day. These data are used for commissioning and detector characterization ("detchar") studies. They are stored in the Cascina circular buffer. The raw data taken during Science Run are copied to the two CCs with tape storages. The comissioning raw data are lost after the depth of the circular buffer, except the part that have been backed-up for longer time as "Interesting segments". For commissioning periods, the interest of copying ∼ 5 minutes of raw data per day to the CCs is being discussed.

2.2.4 AdV Reduced Data Set (RDS). Local. Exported

This data is created with a subset of channels from the raw data, and in general decimated to lower sampling frequency. These are channels which are considered of primary importance for DA issues, to ease the management, access and storage requirements. Additionally, g.w. h(t) channels, at different sampling frequencies, the trend data (see next) channel and the "status flag" channel are stored here. The estimated data flow is 30 GB/day and thus the needed storage for 1 year is 11 TB. This is the data set which, in addition to the h(t) data, rapidly transferred for low-latency scientific searches, will be transferred to aLIGO (as agreed during the negotiations of the new LV MOU). These data are stored in the Cascina buffer (for one year) and copied to al least two AdV CCs.

2.2.5 aLIGO RDS data. Exported

The aLIGO RDS data, a collection of significative raw channels from the two aLIGO detectors, have a flux of 60 GB/day. These data are copied directly from aLIGO CCs to our CCs.

2.2.6 aLIGO h(t) and status flags data. Exported

The aLIGO h(t) files have a flux of 15 GB/day. These data are copied from aLIGO CCs to our CCs. The h(t) and status flags channels are included also in the RDS files, but, given the small size of these files and the need for scientific pipelines to rapidly access to them, we have chosen to copy also these files to our CCs.

2.2.7 Trend data. Local. Exported

In order to quickly visualize the interferometer signal variations over long time periods (weeks or months), the minimum, maximum, mean value and rms of every fast channels is computed every second and stored in the trend data stream. The trend frame builder server receives the whole data stream from a dedicated consumer running on the main acquisition line. It computes the trend data for each raw channel present in the frame. It also contains statistical information about the DAQ system. The trend data frames will cover at least 30 minutes. The estimated data flow is 4 GB/day.

2.2.8 Minute Trend data. Local. Exported

In addition to the previous data , minute trend data are stored and used to study time evolutions over longer periods. It would represent of the order of 1/60 of the trend data flow. These data are used for commissioning and detector characterization studies.

2.2.9 AdV h(t) and status flag data. Local. Exported

To allow faster data analysis processing, the h(t) stream is also stored on disk as a separate file. The file contains h(t) channels at different sampling rates, and one channel with "status flags" (Science, Lock, Injection flags). The h(t) data represent a flow of 7 GB/day. Thus the needed storage for 1 year is 2.6 TB.

Additionally, for the science data, the h(t) time series can be reprocessed off-line when the calibration parameters are better known.

These data are used for online (low latency) GW searches (read in the shared memory in this case) and in the CCs for offline analysis.

These data are stored in the Cascina buffer and copied to all the AdV CCs.

2.2.10 aLIGO h(t) and status flag data. Local.

The aLIGO h(t) data stream from H1 and L1 contain both the g.w. strain channel and one channel with the status flags. They represent about 15 GB/day. The data are used in Cascina for online (low latency) g.w. searches.

2.2.11 Mock Data Challenges (MDC) h(t) frames. CC

To perform tests and comparison of different pipelines it will also be important to work with h(t) frame files to which signals have been added in software. These data are different for the different science groups and can be generated at AdV CCs or at aLIGO clusters and then copied to AdV CCs. 3 TB/yr are needed for each science group (CW, CBC,Burst,Stochastic). The total storage needed is thus 12 TB/yr.

2.2.12 Summary table for the IGWD data set

Next two tables, table 2.1 and table 2.2, summarize the characteristics of this primary data set. The data flow and disk space are our best estimations at today. For the data stored in a circular buffer at Cascina, the planned buffer length is given, with the associated estimation of disk space. Additional space can be needed for interesting segments of data to be stored for longer time.

Data	Data flow [GB/day]	Buffer length in Cascina [year]	Buffer space in Cascina [TB]	Offline storage/year [TB]
Online frames	—	–	–	–
Full Bandwidth raw	4000	0.008	12	–
Raw data	2000	0.5	385+few 10's	745
AdV RDS	30	1	11	11
aLIGO RDS	60	0	0	22
aLIGO h(t) and status flags	15	0	0	6
Trend data	4	3	5	1.5
Minute trend data	0.07	3	0.1	0.025
AdV h(t) and status flags	7.5	1	3	3
aLIGO h(t) and status flags	15	0.5	3	–
MDC h(t)	–	–	–	12
Total	6215	–	420	801

Table 2.1: Summary Table for the IGWD data. Offline storage space does not include multiple copies of the data. Total values are rounded up. "Online frames" are data read directly from DAQ shared memory.

Data	Input	Output	Features
Online frames	DAQ shared memory	online processing	online data, not stored Used for low-latency processes
Full Bandwidth raw	DAQ	DAQ, commissioning	DAQ front-end debugging and full bandwidth. L.
Raw data	DAQ and online processing	inputs for commiss., detchar	channel decimation from F.B. raw. L. E.
AdV RDS	raw data plus online processing	inputs for detchar and science analysis	selected channels from raw data and online processing. L. E.
aLIGO RDS	Data transfer from aLIGO CCs	input to detchar and science analysis	reduced data set of aLIGO data. E.
aLIGO h(t) and status flags	Data transfer from aLIGO CCs	input to detchar and science analysis	the main g.w. aLIGO channel. E.
Trend data	raw data plus online proc.	web monitors, RDS files	1 second statistical properties. L. E.
Minute Trend data	Trend data	web monitors RDS files	1 min statistical properties. L. E.
AdV h(t) and status flags	raw data plus online proc.	inputs to detchar and to science analysis	the main g.w. AdV channel and status flags. L. E.
aLIGO h(t) and status flags	Data transfer from aLIGO CCs	input to detchar and science analysis	the main g.w. aLIGO channel and status flag. L.
MDC h(t) frames	h(t) frames with injections	input to MDCs	Tests and comparisons Science pipelines. CC

Table 2.2: Summary table for the IGWD data. L. = data in the local circular buffer or storage. E. = data exported to one or both CCs. CC = data created in the CCs.

2.3 Commissioning and Detector characterization data

We describe here the data sets used for commissioning and detector characterization studies and the data sets produced by these studies. These data are analyzed and produced at Cascina by online, in-time and off-line processes. They are needed at Cascina for online analysis, online detector characterization, data quality estimation, calibration, noise monitoring. Some of these data are also used for off-line searches.

Some processes also build frame files while others have formats with different outputs (ROOT files, text files, databases, web pages).

2.3.1 Commissioning and calibration

2.3.1.1 Interesting data segments (DS). Local

Time segments of the streams described in previous section can be of particular interest for commissioning and detector characterization. Such selected "data segments" could be stored at Cascina for longer periods than the standard buffer lengths, on a disk space created for this purpose. Their presence on disk will not be limited in time but by the available storage allocated to them. These data might be, on the basis of actual needs, calibration data, science g.w. data, astro-watch data. The size needed to store these data is 30 TB, which is a reasonable choice on the basis of our experience in Virgo. We plan to define, as "DS":

- raw data for calibration (100 hours/year), which amounts to 10 TB to keep the data from last year;
- raw data for interesting commissioning periods (of the order of few 10's of hours/year), which amount to few TB;
- trend data for interesting commissioning periods or Science Run periods (few months of data), which amounts to roughly 1 TB;
- RDS data for interesting commissioning periods or Science Run periods (few months of data), which amounts to 10 TB;
- h(t) data for Science Run periods (few months of data), which amounts to 3 TB;
- "other" data streams, typically not demanding much disk space, with a few more TB if needed.

2.3.1.2 Calibration data. Local. Exported to Lyon

Some calibration processes are run in-time and other are run off-line after the measurements. Most of the processing could be run either at Cascina or in computing centers. However, some output being used online, we have chosen to run the calibration at Cascina. As a consequence, the raw data corresponding to calibration measurements have to be stored in Cascina for about 1 year, and in the computing centers without time limitation. There are three different types of calibration data:

- raw data, 8 TB/yr. These are part of the "inteesting data segments".
- frame files from calibration processes ($\approx$ 2 TB for two years).
- ROOT files and web pages, increasing by $\approx$ 10 GB/year, permanent storage in a backed-up zone.

These raw data and frame files are analyzed to get the final calibration product : the mirror actuator calibration parameters, stored in ROOT files.

The 1 TB/year frame files contain the new processed channels, but also a selection of channels that are already in the raw data. These frames are clearly redundant, in particular this is information which might be stored in the interesting DS. But given it is a small storage request, we prefer to leave to commissioners the possibility to use these data.

The calibration raw data are expected to represent of the order of 100 hours per year, hence 8 TB of raw data per year. In addition to these there is the need to store 2 TB of processed calibration frame data, which cover 2 years.

2.3.2 Summary table for commissioning and calibration data

Next two tables, table 2.3 and table 2.4, summarize the characteristics of the calibration data output.

.

Data	Data flow [GB/day]	Buffer length in Cascina [year]	Buffer space in Cascina [TB]	Offline storage/year [TB]
Calibration output	2.7	space limited	1 TB (+10GB/yr)	1 TB (+10 GB/yr)
DS	83	space limited	30	–
Total	85.7	–	31	1

Table 2.3: Summary Table for commissioning and calibration data. Offline storage space does not include multiple copies of the data.

Data	Input	Output	Features
Calibration output	raw data	ROOT files with the final calibration	Output of calibration processes. (frames, ROOT files, web pages) L. E. to CC1
DS	raw data, trend data, RDS, h(t)	input for commissioning and detchar studies	selected segments data. L.

Table 2.4: Commissioning and calibration data characteristics.

2.3.3 Detector characterization: data quality data

2.3.3.1 Omicron Triggers data. Local. Exported to CC1

Omicron triggers data are stored as ROOT files. These data are used by the online vetoes production pipelines. Needed storage for 1 year is about 2 TB.

2.3.3.2 Online vetoes production data. Local

These pipelines (UPV, Excavator) use the Omicron Triggers data, and the raw frame files (Excavator only) to produce DQ segments, stored in DQSEGDB and to be used by online and offline analysis. Those pipelines have an offline part running in-time to provide input parameters for the online part which produces the DQ segments. Needed storage for 1 year is less than 2 GB for online vetoes stored in DQSEGDB and a few tens of GB for intermediate results produced by the offline part of the pipelines. In addition, the DMS will produce online vetoes. It will requires CPU power but no data storage.

2.3.3.3 DQ segments. Local. Exported

These are the segments with DQ information. They are obtained with a query to DQSEGDB and might be stored temporarily as simple text files. Those segments are used as input to the off-line analysis (mainly CBC and Burst) to reject background events. The content of DQSEGDB is supposed to grow by about 20 millions of DQ segments per year (online segments and reprocessed segment), which represents about 2 GB per year.

2.3.3.4 Spectrogram data. Local. Exported to CC1

The spectra production takes as input the online raw data stream. The output spectra are stored in frame files (spectro data stream) stored in a dedicated disk area. Those files are used to produce spectrogram plots displayed in the MonitoringWeb framework. The needed storage for 1 year is 1 TB for the spectro data stream and 100 GB for the archived spectrogram plots.

2.3.3.5 MonitoringWeb data. Local. Exported to CC1

The archive files of the MonitoringWeb framework represent about 0.8 TB/year (including the 0.1 TB/year to store the spectrograms plots). They should be kept permanently at Cascina, in a backed-up zone and/or exported to CC1 to avoid long reprocessing of the plots in case of any loss.

2.3.3.6 DQ developments data. Local. Exported to CC1

Some storage is needed to test new DQ developments. This work is done off-line in CC1 or in Cascina and needs Omicron triggers and raw data files. The outcome of these studies is new online vetoes or additional information for glitch studies and glitchness reduction. About 0.5 TB/year are needed to store temporary results.

2.3.4 Summary table for detector characterization: data quality (DQ)

Tables 2.5 and 2.6 summarize the sizes and characteristics of the data used and produced for DQ studies.

Data	Data flow [GB/day]	Buffer length in Cascina [year]	Buffer space in Cascina [TB]	Offline storage/year [TB]
Omicron triggers	5.4	2	4	2
Online vetoes production data	0.01	permanent	negligible	negligible
DQ segments	–	–	negligible	negligible
Spectrogram data	3	2	2	1
MonitoringWeb data	2	permanent	0.8	0.8
DQ developments data	1.3	2	1	0.5
Total	11.7	–	7.8	4.3

Table 2.5: Summary Table for data quality storage needs. Again, offline storage does not include multiple copies of the data.

2.3.5 Detector characterization: Noise studies data

Noise Monitors (NM) are a set of pipelines designed to monitor noise characteristics and evolution and produce summary reports and data for offline analysis. NM archive the results in dedicated MySQL databases in Cascina and also in other file formats. They are implemented under a common framework called Noise Monitors Application Programming Interface (NMAPI). NMAPI provides a common interface to present the NM reports and a web GUI to access the output data of the NMs.

Data	Input	Output	Features
Omicron triggers	online frames	ROOT files and entries in DQSEGDB	DQ veto production (used by UPV, Excavator) L. E. to CC1
Online vetoes production data	Omicron triggers, raw data	entries in DQSEGDB	DQ segments for science offline analysis. L.
DQ segments	DQSEGDB	DQ time segments	files used by offline pipelines L. E.
Spectrogram data	raw data	Spectra stored in frames plots for MonitoringWeb	L. E. to CC1
MonitoringWeb data	Informative data from pipelines	Plots on Web	To display information and plots L. E. to CC1
DQ developments, tests	Omicron triggers raw frames	results of tests	investigation studies L. and CC1

Table 2.6: Summary table for data quality.

2.3.5.1 NoEMi data. Local. Exported.

NoEMi reads the raw frame files. The output are list of lines, which are inserted in the MySQL "Lines DB", summary reports, SFDB and Peakmap files for offline analysis. The storage requirements are 12 TB/year for the peakmaps and a negligible amount of space (few MB/yr) for Lines DataBase, assuming to analyze 100 auxiliary channels. The 2 TB/year needed to store the SFDB files for the CW searches have been reported in Section 2.4 and are exported to CC2 and CC*.

2.3.5.2 SILeNTe data. Local

SFOS reads raw frame files and produces plots and text files. The required disk storage is negligible. Most of the plots could be done on fly.

2.3.5.3 Regression monitoring data. Local

The "Bilinear coupling" pipeline reads raw frame files. The output are text files or ROOT files. To monitor 30 channels a disk storage of 12 GB/day is required.

2.3.5.4 WDF data. Local

WDF runs either in online mode, reading online frames or in in-time mode, reading raw frame files from disk. When used as online tool the output are MySQL, "WDF DB" entries and plots. As offline tool it produces ASCII files or plots. Most of the plots will be produced on-the-fly, querying the MySQL "WDF DB". To archive 300 channels, as we plan, we will need 300 MB/day (0.1 TB/yr).

2.3.5.5 Coherence data. Local

The pipeline reads raw frame files and archives results in a MySQL database, “Coherence DB”, which will require 0.2 TB/year.

2.3.5.6 Non stationary monitoring data. Local. Exported

The pipeline reads online frames and produces noise statistics. For the summary report it produces html pages and plots. It gives “trend data” , stored in the trend data frame files (Sect. 2.2). (thus no additional storage is needed in CCs for these data). The required storage is 0.2 TB/year.

2.3.6 Summary table for detector characterization data: Noise studies

Next two tables, table 2.7 and table 2.8, summarize the characteristics of the data for noise studies.

Data	Data flow [GB/day]	Buffer length in Cascina [year]	Buffer space in Cascina [TB]	Offline storage/year [TB]
NoEMi data	33	1	12	12
SILeNTe data	negligible	negligible	negligible	negligible
Regression data	11	0.1	0.4	–
WDF data	0.3	1	0.1	–
NonStatMoni data	0.6	1	0.2	– *
Coherence data	0.6	1	0.2	–
Total	45.5	–	12.9	12

Table 2.7: Summary Table for the storage of detector characterization (noise studies) data. * indicates the needed storage is integrated in the trend data

Data	Input	Output	Features
NoEMi data	raw frames, with status flags	summary web pages, online SFDB and Peakmap files, entries for “lines DB”	Lines: veto for CW, input for bilinear coupling L. E.
SILeNTe data	raw frames	text files and plots	hints for Non linear coupling for glitches and lines. L.
Regression data	raw frames and NoEMi lines	entries to DB, text files, plots	hints for Non linear coupling between lines. L.
WDF data	raw frames or RDS	entries to DB, text files, plots	Triggers rate L.
NonStatMoni data	raw frames data	entries to DB, text files, plots, trend data	Monitor for slow non stationary noise L. E.
Coherence data	raw frames data	entries to DB text files, plots	Lines: correlations between channels. Linked by NoEMi L.

Table 2.8: Summary table for detector characterization (noise studies) data.

2.4 Science Analysis Data

We describe here the data sets produced by the Science analysis. Here we are indexing the CCs, following the scheme at 3.1, namely CC1 is the CC where mainly DETCHAR analysis are run, CC2 the other big CC with tape storage and CC* all the CCs where the scientific analyses are run. To give numbers here use a reference time of 1 yr, that is 1 yr of data taking with 100% duty cycle.

2.4.1 Burst

Table 2.9 and table 2.10, summarize the characteristics of the data sets used and produced in the Burst searches.

Data	Data flow [GB/day]	Buffer length in Cascina [year]	Buffer space in Cascina [TB]	Offline storage/year [TB]
cWB offline	11	–	–	4
cWB online	2.7	–	–	1
cWB pre-conditioning (de-noised h(t) frames)	24.6	–	–	9
Mock Data Challenge (MDC)	2.7	–	–	1
STAMP	x	x	x	x
X-pipeline triggered data	8.2	–	–	3
Total	49.3	–	–	18

Table 2.9: Summary Table for all the BURST searches output data. Numbers are educated guesses based on previous pipeline versions and for offline analyses include the necessary number of re-analyses of data and the tests on simulated data for R&D. We still have no predictions ("x") for the needed storage for the results of the STAMP pipeline.

Data	Input	output	Features
cWB (offline) data	h(t) frames or de-noised h(t) frames or MDC. DQ segments	ROOT and ASCII files, web pages	event candidates, false alarms, detection efficiencies and signal reconstruction. CC2,CC* and aLIGO clusters
cWB (online) data	online h(t) frames, and online status flags frames	input to GraceDB, ROOT and ASCII files, web pages	event candidates, false alarms, detection efficiencies and signal reconstruction aLIGO clusters
cWB pre-conditioning	h(t) frames, raw frames, DQ segments	de-noised h(t) frames. ROOT files	Clean the noise part predictable from environmental noise and instrumental monitoring CC2, CC* and aLIGO clusters
Mock Data Challenge (MDC) output	MDC h(t) frames with signal injections	cWB or burstMDC	Used for for comparison and tests CC2,CC* and aLIGO clusters
STAMP	h(t) frames DQ segments	candidates and false alarm triggers	CC1 and aLIGO clusters
X-pipeline triggered data	h(t) frames, DQ segments,	Matlab files with results	event candidates, false alarms, detection efficiencies, signal reconstruction CC1 and aLIGO clusters

Table 2.10: BURST searches: summary of the data and their characteristics

2.4.2 CBC

Table 2.11 and table 2.12, summarize the characteristics of the data sets used and produced in the CBC searches.

Data	Data flow [GB/day]	Buffer length in Cascina [year]	Buffer space in Cascina [TB]	Offline storage/year [TB]
ihope and GWtools output data	10.9	–	–	4
LALinference, TIGER output data	1.4	–	–	0.5
MBTA output data	1.4	1	0.5	0.5
Total	13.7	–	0.5	5

Table 2.11: Summary Table for all the CBC searches output data

Data	Input	output	Features
ihope and GWtools output data	h(t) frames, DQ segments	Results: triggers, xml files for template banks, for all the detectors	main CBC pipeline results At CC2,CC* and aLIGO clusters
LALinference output data	ihope and GWtools output data	output results	Parameter estimations At CC2,CC* and aLIGO clusters
MBTA output data	online h(t) frames and with state vectors and frame files	triggers to GraceDB	low-latency triggers At Cascina and CC1

Table 2.12: CBC searches: summary of the data and their characteristics

2.4.3 CW

Table 2.13 and table 2.14, summarize the characteristics of the data sets used and produced in the CW searches.

Data	Data flow [GB/day]	Buffer length in Cascina [year]	Buffer space in Cascina [TB]	Offline Storage/yr [TB]
Ephemerides	negligible	–	–	negligible
SFDB (PSS search)	5.5	1/12	0.17 *	2
Peakmaps (PSS All-Sky)	2.7	1/12	0.08 *	1
Candidates (PSS All-Sky)	1.4	–	–	0.5
Follow-up peakmaps (PSS All-Sky)	5.5	–	–	2
Follow-up results (PSS All-Sky)	0.3	–	–	0.1
SBL (PSS Targeted)	0.04	–	–	0.015
Corrected time series (PSS)	0.3	–	–	0.1
2-days segments (Polgraw All-Sky)	1.4	–	–	0.5
Candidates (Polgraw All-Sky)	49	–	–	18
Heterodyne Glasgow data	0.04	–	–	0.015
Search results (Polgraw coherent)	0.3	–	–	0.1 (100 MB for 1 pulsar)
Directed search out spectra	1.2	–	–	0.45
Polynomial search out data	600	–	–	1
Total	68.5	–	0.25	25

Table 2.13: Summary Table for all the CW searches output data. For targeted searches 3 detectors and O(100) targets are considered. * indicates data produced by NoEMi at Cascina, exported (at most) every month and deleted in Cascina.

Data	Input	output	Features
Ephemerides	JPL data PSS code	Tables or vectors for SFDB data	CC2,CC* (Rome, Pisa)
SFDB (PSS All-Sky)	h(t) frames with status flag data, Ephemerides	Peakmaps (PSS All-Sky)	FFT data base Cascina (NoEMi), CC2, CC* (Rome)
Peakmaps (PSS All-Sky)	SFDB (PSS All-Sky)	Candidates (PSS All-Sky)	Time/frequency peakmaps Cascina (NoEMi), CC2,CC* (Rome)
Candidates (PSS All-Sky)	Peakmaps (PSS All-Sky)	Follow-up peakmaps (PSS All-Sky)	parameters of the candidates CC2, CC* (Rome and Budapest)
Follow-up peakmaps (PSS All-Sky)	Candidates (PSS All-Sky)	Follow-up results (PSS All-Sky)	time/frequency refined peakmaps CC2, CC* (Rome)
Follow-up results (PSS All-Sky)	Follow-up peakmaps	Candidates (final result)	parameters of the final candidates CC2,CC* (Rome, Budapest)
SBL (PSS Targeted)	SFDB (PSS)	Corrected time series (PSS targeted)	Band extracted time/frequency data CC2, CC*, (Rome)
Corrected time series (PSS Targeted)	SBL (PSS Targeted)	upper limit/signal parameters	Final down-sampled data CC2, CC* (Rome)
2-days segments (Polgraw All-Sky)	SFDB (PSS)	Candidates (Polgraw All-Sky)	Input time data in a small band CC2, CC* (Polgraw)
Candidates (Polgraw All-Sky)	2 days segments	Candidates or coincidences between candidates	Candidate parameters CC2, CC* (Polgraw)
Heterodyne Glasgow data	Copied from LSC clusters	Search results (Polgraw coherent)	CC2, CC*(Polgraw)
Search results (Polgraw coherent)	Heterodyne-d Glasgow data	Results	upper limit or set of parameters CC2,CC* (Polgraw)
Directed search out spectra	h(t), DQ segments, Ephemerides,	Results	output spectra CC2,CC* (Pisa)
Polynomial out data	h(t), SFTs , Ephemerides,	polynomial coefficients, correlation factors	output parameters files CC* (Nikhef)

Table 2.14: CW searches: summary of the input and output data and their characteristics. In brakets we indicate the CC* where these pipelines are currently running, besides CC1

2.4.4 Stochastic

Data 2.15 and table 2.16, summarize the characteristics of the data sets used and produced in the Stochastic searches.

Data	Data flow [GB/day]	Buffer length in Cascina [year]	Buffer space in Cascina [TB]	Offline storage/year [TB]
Isotropic output (3 pair)	negligible	–	–	negligible
Spherical Harmonic output (3 pair)	9.8	–	–	3.6
Total	9.8	–	–	3.6

Table 2.15: Summary Table for all the STOCHASTIC searches output data. Here we have assumed 3 pairs of detectors and four spectra.

Data	Input	output	Features
Isotropic data	h(t) frames with status flags for all detectors	text result files	cross-correlation, statistical parameters At CC1 and Nice farms
Spherical Harmonic data	h(t) frames with status flags for all detectors	text and mat result files	cross-correlation, statistical parameters At CC1 and Nice farms

Table 2.16: STOCHASTIC searches: summary of the data and their characteristics

2.4.5 Summary table for all the Science Analysis Data

Table 2.17 contains the summary of all the data used and produced by the Science searches.

Data	Data flow [GB/day]	Buffer length in Cascina [year]	Buffer space in Cascina [TB]	Offline storage/year [TB]
BURST	50	–	–	18
CBC	13.7	1	0.5	5
CW	68.5	–	–	25
STOCHASTIC	9.8	–	–	3.6
Total	142	1	0.5	51.6

Table 2.17: Summary Table for all the science analysis output data

2.5 Summary tables for all data

We report here, in Table 2.18 the summary of the information taken for Tables 2.1, 2.3, 2.5, 2.7 and 2.17.

As usual, here offline storage space does not include multiple copies of the data.

Data	Data flow [GB/day]	Buffer length in Cascina [year]	Buffer space in Cascina [TB]	Offline storage/year [TB]
IGWD data	6215	(0.008-3)	420	801 *
Calibration, DS	85.7	permanent	31	1
Detchar: DQ	11.7	1	4.3	4.3
Detchar: Noise	45.3	(0.1-1)	12.9	12
Science analysis	142	1	0.5	51.6
Total	6485	(0.008-permanent)	469	870 *

Table 2.18: Summary Table for all data. Offline storage space does not include multiple copies of the data.*: this is the storage needed for a run of one year, with the requirement that we keep in Cascina only 6 months of raw data

Part III

AdV Data management, distribution and access

Chapter 3

Data management and distribution

3.1 Basic data management and distribution rules

To explore gravitational wave physics with the Advanced Virgo detector the Collaboration has defined a Computing Model that fully supports *accessing* and *analyzing* the data.

Figure 3.1 gives a big picture of the data workflow for what concerns scientific data analysis (DA) and detector characterization (Detchar) activities in AdV.

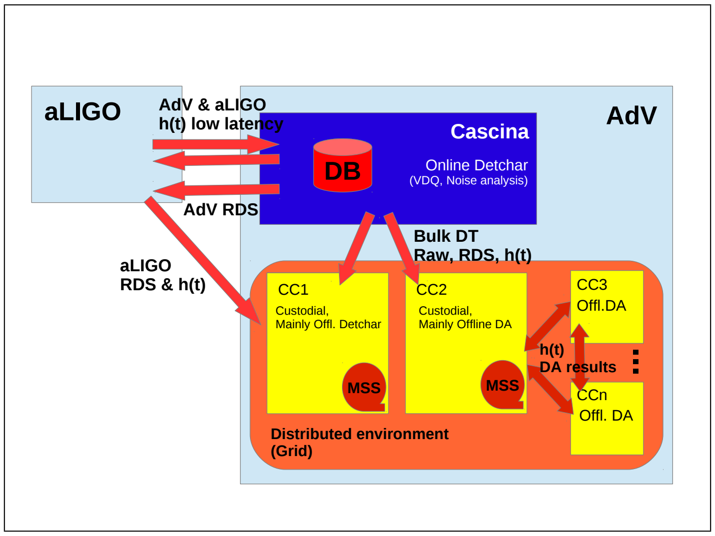

Figure 3.1: Data workflow for DA and Detchar activities in AdV.

Different kinds of data, as shown in Fig.1.1, are produced by the AdV detector, firstly stored at the EGO site in Cascina and copied to AdV and aLIGO CCs, following these basic considerations, based on our experience with the Virgo detector:

- Data which cannot be (easily) reproduced - where easily means intensive CPU work or expensive human activity - have to be stored at least in two external CCs In particular, the raw, RDS and h(t) data need to be copied to a minimum of two CCs.
- These data are processed by online and in-time applications and stored at EGO for a typical period of 6 month. No permanent backup at EGO is currently foreseen, except backup of selected data for crash recovery.

- the most CPU consuming pipelines will run in a distributed computing environment (GRID/CLOUD) which will gather the CPU resources of all the CCs accessible by the AdV organization. In order to optimize data access efficiency we envisage to replicate the intermediate data sets of the pipelines in all the involved CCs. The additional storage requirement for this is small compared to the full data set, as these analyses do not use the raw data.

- All the acquired and commissioning data, astrowatch and scientific runs will be stored on **tapes**, which implies the yearly increase of tape capacity by an amount which will depend on the run time of the detector.

- Data of the current run will be stored on disk. At regime, when the detector will take data continuously for 1 year, this requires 1 PB of disk storage for each of the two CCs which will store the raw data also.

3.1.1 Data management at EGO-Cascina

The Cascina facility is dedicated during the runs to data production and to detector characterization and commissioning analysis, which have the need to run "on-line" (with a very short latency, from seconds to minutes, to give rapid information on the quality of the data) or "in-time" (with a higher latency, even hours, but which again produce information on the quality of the data within a well defined time scale). The detector characterization analyses give support to both commissioning and science analysis. The primary data set, described in section 2.2, and the commissioning and detector characterization output data, described in section 2.4, are produced at the EGO-Cascina site. The production rate is continuous, both during the science mode periods and the commissioning periods. To insure the fulfillment of the workflows described in Part I, AdV places the following targets for the management and storage of these data. The main data streams are stored and accessible for a given time period in Cascina for commissioning and detector characterization. Their backup is done transferring them to the AdV CCs where they are archived and accessible for offline analysis.

The main data streams must be available for the commissioning and detector characterization in Cascina circular buffers for the time periods indicated in the table 2.1 (from 3 days to 3 years depending on the streams). They are readable through "frame file list" (ffl) files generated during the storage processing such that the hardware location of the data is transparent for the users.

Interesting "data segments" for commissioning and detector characterization are stored on disk. Their storage is not limited in time, but limited by the available storage allocated to them. These data are readable through the same ffl files as for the circular buffers.

All these data are backed-up for the corresponding periods at Cascina.

3.1.2 Data transfer to the CCs: requirements

The Virgo data are transferred to the CCs for both archiving and access for offline analysis.
The *data transfer framework* includes the data transfer itself, the check of the data integrity and the ffl file generation for local data access in the CCs. A procedure for long term integrity checks will also be in place by ADE (details in the Implementation Plan). Besides this, we require:

- to check the data consistency in the local buffer, before distributing them, in order to prevent to distribute bad data. For examples: are there missing frames, are there missing channels, ...?) . This might be checked before flagging that the data are ready to be transferred.

- to provide a DT monitoring web page.

In the following, we list the rules specific to the "low-latency" and "bulk" DTs.

3.1.2.1 Bulk data transfer

The latency required for standard data transfer varies according to the data type as follows:

- AdV raw data: 1 day maximum (from Cascina to AdV CCs)
- AdV RDS and trend data: 1 day maximum (from Cascina to AdV CCs)
- AdV h(t) data: 1 day maximum (from Cascina to AdV CCs)
- aLIGO RDS data: 1 day maximum (from one aLIGO cluster to AdV CCs)
- aLIGO h(t) data: 1 day maximum (from one aLIGO cluster to AdV CCs)
- AdV RDS data: 1 day maximum (from Cascina to one aLIGO cluster)

All the data streams are transferred during Science Runs, Astro-watch and Calibration periods. During commissioning periods, the Virgo RDS and trend data are permanently transferred, while only $\sim$1% of raw data are transferred (5 minutes per day). The segments of raw data corresponding to calibration data are also transferred.

3.1.2.2 Low latency data transfer

The low latency data transfer is needed for online analysis running at Cascina or in LSC cluster. The transfer must be done within few tenth of seconds for the following data:

- aLIGO h(t) data: from one aLIGO cluster to Cascina
- Virgo h(t) data: from Cascina to one aLIGO cluster

3.1.3 Data Transfer to CCs: description

3.1.3.1 Bulk Data Transfer (DT)

To ensure the continuous data distribution to the different data centers the bulk data transfer system is modular and capable to adapt to the different storage systems at the endpoints using different transfer protocols (currently iRODS and GRID/lcg) and selecting different data types.

This is an important requirement for the DT architecture.

Nevertheless, we would have in place by ADE a Bulk Data Transfer system which uses the same protocol at least for the two main CCs on which we rely today, to minimize the overall complexity of having too many different modules.

The bulk data transfer system is coupled with the data handling system at Cascina in order to provide an automatic sequential transfer. By ADE we will have in place an interface for the update of the Locator Database described in section 4.1.1.

The topology is star-shaped with Cascina at the center and the CCs at the endpoints.

No provision is made to manage third-party transfers automatically inside the system, unless we will manage to use the same protocol for DT towards both the CCs, in which case the third-party transfer will certainly become an appealing solution.

In both cases, the outcome of DT will be automatically inserted in the Location Database. The bulk data transfer system queues the files asynchronously preserving the time ordering, this allows the feeding of possible pipelines at the endpoints for the processing or conversion of the files in another format, with the drawback of suspending the transfer in case of the transmission failure of the head of the queue and a non optimal performance for the smallest files. Data integrity is guaranteed by the underlying transfer protocol, but the whole data integrity problem should be better handled asynchronously in the framework of the Data Location service in order to check the file in the final location periodically to catch the displacements due to wanted or unwanted reorganizations at the endpoints.

A web monitoring of the DT status is also required.

The data transfer needed to distribute aLIGO RDS and h(t) data to AdV CCs or to the Cascina farm is currently part of the Bulk DT.

Given the fact that these data are only used to do off-line analysis, there is no real need to use Cascina as a bridge to distribute them, which is in any case one possible scenario, and we can probably distribute them directly to AdV CCs. Thus, we have to possible solutions:

- copy aLIGO RDS and h(t) data to our CCs through Cascina. This solution is already working, as it has been succesfully used to tranfer the old LIGO data. This is under the EGO responsability
- copy aLIGO RDS and h(t) data directly to our CCs. This solution is completely new, and the responsability has to be still assigned. Technical solutions need to be exploited, some issues need to be agreed with our aLIGO colleagues, and finally tested.

The "Implementation plan" gives details and the status of the work and decisions on the two possible solutions.

3.1.3.2 Low latency data transfer

A DT system, separated form the Bulk DT, is needed to guarantee the success of "low-latency" searches. In fact, as reported in Section 3.1, the needed latency is so small (seconds) that, in this case, we can't follow the same basic rules of the Bulk Data Transfer.

The data to be copied is the g.w. channel, h(t) including status vectors, from LIGO to Cascina (for "Low latency searches" done in Cascina) and from Cascina to LIGO (for "Low-latency searches" done on LIGO clusters).

Besides this, there is the need for a rapid data transfer to LIGO, also of the outcomes of the search done in Cascina, which are a very few data, in the format of frame file triggers and entries in the remote data base (the actual solution is called "GraceDB").

The actual solution, which fulfills the few-seconds latency requirement, is to transfer the data using the "Control model" ("Cm") advanced file transfer, and to read the g.w. h(t) data in Cascina directly from the DAQ shared memory.

3.1.4 Data management at AdV CCs

All the scientific analyses, except the low-latency searches, are carried on outside the EGO-Cascina environment. The AdV CCs receive a copy of the data and provide storage resources for permanent data archiving. They must guarantee fast data access and computing resources for off-line analyses. They must also provide the network links to the other computing resources.

Given the fact that, as clarified in the tables in Section 8.7, the CPU requests is going to become larger and larger, to cover increasingly needs for the parameter space of these searches, we strongly support the possibility to use more computing centers, in particular aiming to add computing resources under GRID/CLOUD.

Another very important need for AdV is to increase the availability and usability of the computing resources and enable the usage of a Grid-enabled, LIGO-compatible Condor cluster for Virgo people, having also local submission capability.

Of course, the various computing and storage tasks have to be coordinatedly distributed over these CCs. To this goal, we need:

- to guarantee, in at least one CC where the raw data are archived, local access to the data and to the computing resources, as requested for the offline detector characterization studies and for software development and testing purposes;
- that the CCs where the DA offline analyses will run, will be integrated in a homogeneous distributed environment, to fully exploit the available computing power;

- in order to fulfill the request in the LV agreement about sharing of resources[4] DA pipelines must be able to run in the different environments of AdV and aLIGO CCs (European GRID and LDG) It is also important to develop a software layer to allow a transparent data access to the storage resources.

3.1.4.1 Replication of DA data sets

As written in the introduction to this chapter, we envisage to develop a software framework to replicate the intermediate stage files of the DA pipelines to all the CCs involved in the analysis. This proposal has not yet been discussed within the collaboration.

Chapter 4

Data Access Model

We describe here the Data Access model for AdV. The model applies only to the principal sequences of data, in particular the raw data set, the RDS data and the g.w. science channel h(t) for all the detectors of the network. These data are all stored in "frame" format files, the same for all the interferometers of the network, described in Chapt. 7.

The local access to the files is guaranteed using text catalog files, called the "ffl" ("frame file list").

There is no Data Access Model for the outcomes of the scientific pipelines, as here the huge variety of the existings and foreseen pipelines is such that each Science group has found its own solutions. This is also due to the fact that usually the outcomes of the analyses are stored in files of relatively small dimensions and thus easily managed locally by the users.

We envise the need to organize the Data Access Model for the outcomes of the science analyses in view of a possible release of triggers and data to the public. But this is actually a so far perspective that we have not considered it here. We will add a specific section to this Model when it will be needed.

With the above constraints, the goals of the AdV data access model can be summarized in one sentence:

Provide the most transparent possible access to the needed data by the Advanced Virgo community users irrespectively of the diversity of the data centers where the files are placed.

An additional condition, imposed by the fact that AdV computing and storage resources are spread in different administration environments, is that the data access system would need the smallest possible "footprint" in term of requirements from the computing centers. In the model we separate the "catalog/bookkeeping" task from the end-user physical "transparent data access" task.

We aim at having both tasks in place for the first run of AdV.

The implementation schedule for this work is detailed in the "Implementation Plan".

The goal of the first task is to provide a unified catalog of the data distributed among a variety of resources, with an interface to the user giving the expected information.

The goal of the second task is to give a transparent data access, and the data access layer re-worked trying to make the access truly transparent and homogeneous.

4.1 Data Bookkeping

In the AdV data access model we guarantee, for all those users who will want to use it, an entry point for offline computations attached to the Ligo-Virgo Data Quality Segments Database (DQSEGDB), which is the DB for Data Quality (DQ) segments, where the user will be able to browse and select (interactively or via command-line) the main scientific characteristics of the time periods to be analyzed. Given that the project is for offline DA pipelines, there are no important contraints to the latency (e.g. 15 minutes will be enough) needed to have the DQ segments ready. The Locator Service (LS) and the associated Locator Database (LDB), will provide to interactive users and software applications the file locations and characteristics for all the data of shared use present in all the computing centers.

4.1.1 File Locator Data Base

4.1.1.1 Preamble

The AdV Data Analysis will follow an open approach in the selection of the computing and storage resources where to run on. The DA jobs will run in a great variety of environments: the national computing centers at Bologna and Lyon, on LSC clusters, in the Virgo laboratories and in Cascina. All these resources use very different technical solutions in the interface seen by the Virgo applications both to the computing farms and to the storage systems. In particular one of the biggest problems that users must tackle while moving to different computing centers is finding the filenames and file locations of interest and accessing them.

For example, referring to the situation we have now,one may need to access data through iRODS/Xrootd in one place, via GRID in another and through a POSIX filesystem elsewhere, provided that a text file with the list of the files of interest is produced.

4.1.1.2 The project

The "Locator Service" (LS) and "Locator Database" (LDB) will deal with the file locations in all the supported storage systems and will provide the lists of files for each of them according to the client requests from each environment. In order for this functionality to be completely transparent to the user the Locator server/s should be complemented by a client library integrated in the AdV applications dealing with the negotiation of the lists with the LS and the selection of the I/O access model suited to the environment where the application is running. This part is postponed to the successive phase when the more general "transparent data access" task will be tackled.

Requirements and functionalities:

- Will be distributed geographically in each computing location in order that the service be available locally in case of unreachability of the central repository
- Will check the consistency of the file layout of every location where shared AdV data are stored
- Will allow the registering/deregistering of data files and replicas, both automatically interfacing with the bulk data transfer and distribution system, and also manually by the users according to a well defined policy
- If possible, it will collect metadata information proper to each storage subsystem, for example in order to know whether a given file in an HSM storage system is staged or not
- Will provide both a GUI interface and a programmatic remote interface
- When it will be integrated with the data access layer it could collect also statistics about the files requests, useful to profile the real data usage

For each storage resource, the LS will have an interface agent that will provide the status of the AdV files on that resource and will check for the consistency; care will be taken that these interfaces are modular and will evolve according to the variation of the related storage resource.

The storage access method to be covered will be different in different CCs (currently iRODS/Xrootd for the resources at CC-IN2P3, POSIX for CNAF and EGO-Cascina, GRID for CNAF and the other CCs).

4.2 Data Access

4.2.1 Local Data Access in the AdV CCs

As already explained, the local access to the files is guaranteed using text catalog files, called the "ffl" (frame file list). Thus, the minimum requirement for the Locator Service is to provide at the "ffl" lists suitable for each computing centers, keeping them uptodate. Due to the (frequent) need of building input data sets from raw data (i.e. building a list of selected channels for a given time period), the read and random access performance from the archiving system is of maximum importance.

The variety of input data sets that change frequently according to the different kinds of analysis makes infeasible to envision a single RDS subset that could substitute the full raw data. Therefore in each center the access to the long term (tape) archiving system is mediated through an on-disk caching buffer capable of storing at least the length of a typical science run (ranging from 6 months to 12 months). This kind of caching is provided by each computing center according to its general purpose architecture (currently, Tivoli TSM for GEMSS at CNAF and Xrootd for HPSS at CC-IN2P3), therefore it is not optimized for the Virgo raw data except for the extent to which a suitable staging policy could be built. This problem is described in the Implementation Plan, possible technical solutions to be tested have been proposed, together with milestones for the process.

4.2.2 Remote Data Access

A truly transparent data access could occur only if the location of the computation is to some extent independent from the data source. This is more true for those spot or one-time accesses that the users need in their own home environment not covered by the Locator Service, or whereas there is not a local copy of the needed data. In this case a "Data Streaming Service" (DSS) is a good solution to provide channel files from rawdata, or other data sets upon requests from users, via a web streaming service or command line. If, in a second phase, a client part were embedded in the Virgo Software Environment the applications could transparently use the service. This component of the full Data Access Model should interface with the caching storage systems at the computing centers, accessing a set of RDS files, and would plug in the Locator Service as a possible source of data. The tecnical solutions for this need to be exploited and they have been described in the Implementation Plan.

Part IV

Software description and management

Chapter 5

Base and Data Analysis software

The sections here have been divided using the same classification in the Part I of the CM (workflows description).

Each subsection gives a short summary of the needs of each pipeline, in terms of characteristics like: OS, method for the analysis (on-line, in-time, off-line), process used to submit it (interactive, batch system, GRID, CLOUD ...), need to use software like e.g. ROOT or commercial software like e.g. Matlab or Mathematica.

Any other relevant computing need or software dependecies which we think important have been also reported here.

We have also listed the names of people who are working on each pipeline and we have given milestones and the final goal for each project, to be able to individuate any crucial path to the goal.

In all those cases where the “SVN Versioning” information is missing, it is because we still have not the versioning on that software. But this is something we plan to fix.

5.1 Commissioning and calibration

The commissioning activities are carried in Cascina computing center since the raw data are only available there, except during Science Runs. The analysis for calibration of the interferometer are also carried in Cascina since some of the results are produced in-time.

Most of the commissioning and calibration analysis are done offline, using data display tools to understand and improve the detector behaviour. The Virgo software dataDisplay is used extensively, and External sofware such as ROOT and Matlab, interfaced with the Frame library for I/O, are also needed.

A more precise estimation of the computing needs for detector operation, commissioning and calibration is being pursued. The software needed by the different AdV sub-systems will also be assessed in this process.

5.2 Detector characterization

Most of detector characterization analysis is done in Cascina computing center, since the results have to be produced in real-time. Some pipeline are linked to online DAQ chain, others read file from disk.

5.2.1 Data Quality

Data Quality activity requires mainly two axis of computing : the online DQ flags production and the tools for monitoring and glitches investigations. One central tool for both axis is the online production of triggers able to provide a reliable view of the glitches that affect the interferometer dark fringe signal.

5.2.1.1 Omicron pipeline

The Omicron pipeline provides triggers, either offline or online, by looking at the dark fringe signal but also to hundreds of auxiliary signals delivered by the interferometer or the environment monitoring. This software requires a minimal set of 40 computing nodes and some disk to store the trigger information produced. The online version of Omicron uses the FdIO library of the Fd package.

5.2.1.2 On-line vetoes

Using a dedicated set of online processes (that use FdIO) or the triggers produced by Omicron, it is foreseen to produce online vetoes with a latency below 1 minute. Some tools will be also dedicated to building the online DQ segments and providing the segments needed by online analyses down to the h(t) frames and to the DQSEGDB database. One computing node is required to run it.

5.2.1.3 Detector Monitoring System (DMS)

The DMS operates in Virgo since 2006 and provides useful information in control room about the status of the interferometer, including the various parts of the DAQ and online processing. It is based on the online production of a set of DQ flags. We plan to upgrade the DMS, to redefine its DQ flags and to use them also for online vetoes production. The DMS uses the Moni and the FdIO libraries and uses a set of php tools to produce the web page shown in control room. Two to four computing nodes are required to run it.

5.2.1.4 MonitoringWeb

This tool provides various information about the interferometer subsystems, online processings or online analyses like MBTA and Omicron. It uses mainly trend data and reaches a latency between 2 minutes and 30 minutes. It is mainly based on bash scripts, ROOT macros and Frame library. It needs about 4 computing nodes and some storage to archive the various plots on disk space visible from the web.

5.2.1.5 Spectrograms

Part of the MonitoringWeb is dedicated to hourly, daily and weekly spectrograms showing various signals useful for commissioning and for monitoring the behaviour of the interferometer's environment. The hourly update of those plots requires at least 16 CPU cores (4 computing nodes). Those spectrogram plots are based on spectro data produced online by the SpectroMoni process which requires one computing node and about 0.5 TB per year to store its output. SpectroMoni uses FdIO library and is connected to the online DAQ stream.

5.2.1.6 DQ developments

The development of new DQ flags or of tools useful to investigate glitches is mainly done using the raw data available on the Cascina site. It is strongly connected to glitch families identification and may occurs several months after data have been taken. This requires raw data to be available for several months at Cascina site. It requires also a dedicated set of computing nodes (at least 4 computing nodes) were can be launched the investigations and the development tests and a dedicated storage area to store temporarily the tests results (about 1 TB).

5.2.2 Summary tables for Data Quality

Table 5.1 gives the status of each pipeline and the people involved in the project in the year 2013 and the goal for the project by Jan. 2015. Table 5.2 gives detailed milestones for the project. Table 5.3 summarizes the main computing features.

Pipeline	SVN Vers.	Responsible and Collaborators	Status(July 2013)	Goal (Jan. 2015)
Omicron pipeline		Robinet, Leroy	Used in ER4, online	Run online in H1,L1,V1. . Provide triggers for low-latency UPV,Excavator, hveto, DQperf, etc.
On-line vetoes		Verkindt, Robinet	Partly used in ER4, online, but needs updates	Provide DQ_VECTOR data and online DQ segments stored in DQSEGDB
DMS		Dattilo, Verkindt, Berni, Hemming,..	Update document to be finalized	Provide online information about detector's state and online DQ flags
Spectrograms		Verkindt	Running	Manage spectro data files more automatically
MonitoringWeb		Verkindt	Running	Include Injections, Data Transfer and Storage information
DQ developments studies		VDQ group	BRMSMon, Excavator, etc... in various states	Ready for use in commissioning

Table 5.1: Summary Table for data quality pipelines needs.

Pipeline: milestones	October 2013	January 2014	May 2014	October 2014	Jan 2015
Omicron pipeline	Running				
On-line vetoes	Requirements defined	Architecture defined	First implementation and tests	Used with DQSEGDB and DQ_VECTOR	
DMS	First draft of upgrade description	All requirements defined	First implementation of flags production	Test connection with online DQ	First implementation of upgraded DMS web pages
Spectro-grams	Running	Improve spectro data storage management	Adapt web pages and web pages ready for commissioning	Spectro data	
Monitoring-Web	Running	Add pages for DQ safety, injections, storage,..	Simplify web page updater	Check for useful plots and commissioning needs	Fully updated MonitoringWeb running
DQ developments studies	Check for needs	Define an online implementation of tools Excavator or UPV	Tests using Env. Monitoring data	First online implementation of a full set of DQ tools	

Table 5.2: Milestones for data quality software

Pipeline	Processors e.g. CPUs/GPUs	Submission method e.g. interactive/batch/ /GRID/Cloud/Dirac	Language/Sofware e.g. C, C++, Matlab,Root	Software dependencies CMT,Fr,FFTw
Omicron pipeline	60 CPU (cores)	online and interactive	C++ and ROOT	Fd, Fr, GWOLLUM
On-line vetoes	4+4 CPU	online and interactive	C++, C, Python	Fd, Fr, GWOLLUM
DMS	4 CPU	online	C, php...	Fd, Fr, ...
Spectrograms	16 CPU	online or in-time	C, bash	Fd, Fr, ROOT
MonitoringWeb	16 CPU	in-time	C, bash, ROOT	Fr, ROOT
DQ developments studies	4 CPU	interactive	C, C++, Python...	Fr, ROOT...

Table 5.3: Main computing features for DQ work

5.2.3 Noise studies

The noise monitor (NM) pipelines are integrated in a common framework NMAPI (as described in the CM). We setup architecture for noise monitoring in such a way to have a single web interface where displaying the results produced by each NM and where it is possible to launch scripts using only the web interface. Each NM relies on its own software enviroment. Most of them needs only free software, integrated in the standard Adv Virgo software environment, others can require the use of commercial software as Matlab.

5.2.3.1 NMAPI

NMAPI will be able to operate using the standard Linux operating systems common to the Virgo public host machines. NMAPI will require the use of a standard web-server, i.e. Apache, IIS. In terms of hardware, no specific requirements outside of the standard configuration are required. NMAPI will be written in PHP. NMAPI will use JavaScript at browser-side, taking advantage of the Jquery library. A MySQL database will be used to store all NM meta information and documentation. NMAPI will also take advantage of the sundry available PHP classes, JavaScript functions and CSS styles already available in the Virgo General collection. These cover areas ranging from website and element formatting to user authentication and dynamic functionality, e.g. form validation. NMAPI will be developed using XHTML 2.01 Obviously, W3C standards will be applied to the UI, while the Web Standards Project (WaSP) will also be used as reference for graphical and UI standards.

Validation will take place using the W3C HTML and XHTML validation service2.

5.2.3.2 NoEMi

The core of the program is a small C executable written using the "Rome PSS software" (see 5.3.6.1) library. It also requires Python with some external modules (matplotlib, numpy, MySQLdb, markup) which provide functionality to analyse, plot, publish and send results to the database. No commercial software is required.

5.2.3.3 SILeNTe

It is coded in Matlab, but it is planned to translate it in C o C++.

5.2.3.4 Regression

It is written in C and relies on root packages. It uses also uses cWB library.

5.2.3.5 WDF

The Wavelet Detection Filter requires the Noise Analysis Package (NAP) library, which is written in C++ and which has python binding. The scripts itself is written in scripting language python. The WDF needs the standard Virgo Common Software enviroment.WDF is under svn and in VCS.

5.2.3.6 Coherence

Coherence is written in python and relies on the NAP interface to raw data. The latest version in still in /virgoDev. (/virgoDev/SisCo)

5.2.3.7 Non stationary monitoring

NonStatMoni is written in C and uses FdIO to access data. The latest version is in /VirgoDeV. /virgoDev/NonStatMoni is the software which analyze the data in real time. /virgoDev/NonStatMoniOffline is the software package to produce offline reports.

5.2.4 Summary tables for Noise studies

Table 5.4 gives the status of each pipeline and the people involved in the project in the year 2013 and the goal for the project by Jan. 2015. Table 5.5 gives detailed milestones for the project. Table 5.6 summarizes the main computing features.

Pipeline	SVN Vers	Responsible and Collaborators	Status of the project (July 2013)	Goal (Jan. 2015)
NoEMi	SVN trunk	A. Colla (Virgo Rome group)	In production	Production of Peak-maps used in the CW analysis; feed Regression NM
SILeNTe	?	F.Piergiovanni, G.M.Guidi	Matlab coded	Integration in NMAPI and C coded
Regression	?	M.Drago	In production	Integration in NMAPI
WDF	svn	E.Cuoco	In production	RDS analysis
NonStatMoni	-	G. Vajente	In production	-
Coherence	-	G.Vajente	In production	-

Table 5.4: Summary Table for Noise pipelines needs.

Pipeline: milestones	October 2013	January 2014	May 2014	October 2014	Jan 2015
NoEMi	-	Produce Peak-maps for CW analysis	DB Re-organisation (see 6.6)	Feed Regression NM	Ready to run on AdV
SILeNTe	-	Integrated in NMAPI	-	C-translation	Ready to run on AdV
Regression	-	Integrated in NMAPI	-	-	Ready to run on AdV
WDF	-	-	Test on RDS	-	Ready to run on AdV
NonStatMoni					
Coherence					

Table 5.5: Milestones for Noise software

Pipeline	Processors e.g. CPUs/GPUs	Submission method e.g. interactive/batch/ /GRID/Cloud/Dirac	Language/Sofware e.g. C, C++, Matlab,Root	Software dependencies CMT,Fr,FFTw
NoEMi	1 CPU per channel for 3 hours/day (34 KHS06*day for 100 channels)	Batch (Cascina)	C, Python	Virgo software
SILeNTe	50 CPU for 12 hours/day	Batch	Matlab	-
Regression	50 CPU for 3 hours/day	Batch	C,Root	cWB library
WDF	50CPU for 24hours	on-line or Batch	C, Python	Virgo software
NonStatMoni	10 CPU for 24hours	on line	C	Virgo software
Coherence	50 CPU for 3hours/day	Batch	C, Python	Virgo software

Table 5.6: Main computing features for Noise work

5.3 Scientific analysis

5.3.1 Low latency searches

5.3.2 Summary tables for low-latency searches

Table 5.7 gives the status of each pipeline and the people involved in the project in the year 2013 and the goal for the project by Jan. 2015. Table 5.8 gives detailed milestones for the project. Table 5.9 summarizes the main computing features.

Pipeline	SVN Vers	Responsible and Collaborators	Status of the project (July 2013)	Goal (Jan. 2015)
CBC low-latency (MBTA)				

Table 5.7: Summary Table for low-latency needs.

Pipeline: milestones	October 2013	January 2014	May 2014	October 2014	Jan 2015
CBC low-latency (MBTA)					

Table 5.8: Milestones for low-latency software

Pipeline	Processors e.g. CPUs/GPUs	Submission method e.g. interactive/batch/ /GRID/Cloud/Dirac	Language/Sofware e.g. C, C++, Matlab,Root	Software dependencies CMT,Fr,FFTw
CBC low-latency (MBTA)				

Table 5.9: Main computing features for low-latency work

5.3.3 Off-line searches

5.3.3.1 Burst

5.3.3.1.1 All-sky all-times offline search using "coherent WaveBurst" pipeline. The pipeline is a joint project LSC-Virgo. cWB (2G) pipeline has been completely restructured with respect to the version (1G) which run over the past observation. The pipeline is under advanced stage of development and the first freezed version will be available by late 2013; at the same time the internal review of the pipeline will start. The overall computational requirements of cWB 2G result similar to those of the previous version, even though the algoritms and internal structure changed significantly. As a general commento, tha computing time neede to perform a search depends crucially by the glitchyness (excess non gaussian noise) of the data. The test of the new version will include the re-analysis of all past observations; a fact that will dominate its 2014 computing request. In 2014 we plan also to start a feasibility study on the transition from CPUs to GPUs for the most computationally demanding modules of the code.

5.3.3.1.2 All-sky all-times offline search for long duration bursts using *STAMP all-sky* pipeline. The pipeline is a joint project LSC-Virgo, originated from the LSC STAMP pipeline for stochastic signal searches. The all-sky search capability is under active development. The main test of the pipeline will be accomplished by analyzing 2007 data, starting from detector pairs. Once the feasibility and the performances will be demonstrated, the pipeline will be reviewed, aiming at a freezed working version in 2015.

5.3.3.1.3 Gamma Ray Burst triggered search using *X-pipeline*. The pipeline is a joint project LSC-Virgo. The current version is considered already final and minor review is needed since the differences with the version used on past data have been kept to a minimum. In 2014 tests of the pipeline will be performed e.g. using Engineering Run 5 and 6.

5.3.4 Summary tables for Burst offline

Table 5.10 gives the status of each pipeline and the people involved in the project in the year 2013 and the goal for the project by Jan. 2015. Table 5.11 gives detailed milestones for the project. Table 5.12 summarizes the main computing features.

Pipeline	SVN Vers	Responsible and Collaborators	Status of the project (July 2013)	Goal (Jan. 2015)
cWB offline	LIGO SVN	Vedovato, Drago Re, Lazzaro with LIGO colleagues	advanced development	freezed version working code review completed past data re-analysis completed
STAMP long bursts		Franco, Bizouard Hello with LIGO colleagues	all-sky capability under development	working version
X-pipeline	LIGO SVN	Was, Leroy with LIGO colleagues	review almost complete past data analysis completed	freezed working version

Table 5.10: Summary Table for Bursts pipelines needs.

Pipeline: milestones	October 2013	January 2014	May 2014	October 2014	Jan 2015
cWB offline	start review	start feasibility study on GPUs	past data re-analysis	review complete	running on new data
STAMP long bursts		start review	past data analysis		
X-pipeline		test on engineering run		test on engineering run	running on new data

Table 5.11: Milestones for Bursts software

Pipeline	Processors e.g. CPUs/GPUs	Submission method e.g. interactive/batch/ /GRID/Cloud/Dirac	Language/Sofware e.g. C, C++, Matlab,Root	Software dependencies CMT,Fr,FFTw
cWB offline	CPUs (study on GPUs)	interactive batch	C++ Root	Fr LAL Healpix (not mandatory)
STAMP long bursts	CPUs	interactive batch	Matlab C++	matapps
X-pipeline	CPUs	interactive batch	Matlab	Fr LAL (not mandatory)

Table 5.12: Main computing features for Bursts work

5.3.4.1 CBC

5.3.4.1.1 Detection of compact binary coalescence signals .

5.3.4.1.2 Extracting parameters, testing GR, and determining the neutron star equation of state with compact binary coalescence detections .

5.3.5 Summary tables for CBC offline

Table 5.13 gives the status of each pipeline and the people involved in the project in the year 2013 and the goal for the project by Jan. 2015. Table 5.14 gives detailed milestones for the project. Table 5.15 summarizes the main computing features.

Pipeline	SVN Vers	Responsible and Collaborators	Status of the project (July 2013)	Goal (Jan. 2015)
ihope, GWTools	git repository git repository	John Veitch Gergely Debreczeni	Being adapted for GRID Prototype apps are running	Ready for analysis Ready for analysys
LALinference, TIGER	git repository git repository	John Veitch Chris Van Den Broeck	Being adapted for GRID Being adapted for GRID	Ready for analysis Ready for analysis

Table 5.13: Summary Table for CBC (offline) pipelines needs.

Pipeline: milestones	October 2013	January 2014	May 2014	October 2014	Jan 2015
ihope,	-	Ready for Virgo CCs	-	Fully tested	Ready for analysis
GWTools	-	Ready for GPUs and CPUs	-	Fully tested	Ready for analysis
LALinference,	Ready for Virgo CCs	-	-	Fully tested	Ready for analysis
TIGER	Ready for Virgo CCs	-	-	Fully tested	Ready for analysis

Table 5.14: Milestones for CBC (offline) software

Pipeline	Processors e.g. CPUs/GPUs	Submission method e.g. interactive/batch/ /GRID/Cloud/Dirac	Language/Sofware e.g. C, C++, Matlab,Root	Software dependencies CMT,Fr,FFTw
ihope,	CPUs	GRID	C, Python	lalsuite dependencies
GWTools	CPU and GPU	Cluster and Grid	OpenCL,C++	lalsuite
LALinference,	CPUs	GRID	C, Python	lalsuite dependencies
TIGER	CPUs	GRID	C, Python	lalsuite dependencies

Table 5.15: Main computing features for CBC (offline) work

5.3.6 CW

5.3.6.1 All-Sky searches for unknown rotating neutron stars

I) Frequency Hough search (Periodic Source Search, PSS) The “Periodic Source Search” (PSS) software is used for both the All-Sky and Targeted searches carried on in the Rome AdV group. It is based on two programming environments: Matlab and C. The first is basically oriented to interactive work, the second to batch or production work. There are also programs developed in Matlab, then compiled by the Matlab compiler and which run on the Grid environment. There is no need to have Matlab on the working nodes, once the code has been compiled. An important part of the package are the simulation modules. There are no constraints on the SL version, the latest stable version in the year 2015 should work. Some work is ongoing, as detailed in the tables, to do the porting towards a possible CLOUD submission (actually following the “DIRAC” project, details in the Implementation Plan). And also some work is ongoing for the porting of the software under GPUs, but here the need for skilled man-power presents an major issue.

II) All Sky F-statistic search The F-statistic all sky search was developed by Polgraw-Virgo group. The main code for this search that analyses 2-day, narrowband time domain sequences is written in C. The narrowband time domain sequences are generated using codes from the PSS pipeline developed by the Rome group. There are also auxiliary codes written in Matlab and Pascal. They are available at Cascina CVS repository at

`https://wwwcascina.virgo.infn.it/cgi-bin/cvsweb/cvsweb.cgi/PolgrawAllSky/`.

There is also work ongoing for the porting of the main code to GPUs.

5.3.6.2 Targeted searches for known pulsars

.

I) "Rome Targeted" PSS pipeline This pipeline takes as input the short FFT database used also for the all-sky search and performs a coherent search targeted to a given known pulsars, for which position and rotational parameters are known with high accuracy. The barycentric and spin-down corrections are done using an efficient time-domain method followed by a down-sampling of the data. It then computes the so-called 5-vectors, i.e. the data and signal templates Fourier components at the five frequencies generated by to the sidereal modulation, and from these a detection statistic used to assess the statistical significance of a given analysis result. The software is written in Matlab but a standalone compiled version has been also produced in preparation of the AdV era when several potential sources will be targeted.

II) Targeted F and G statistic search The targeted F and G statistic search was developed by Polgraw Virgo group. The codes for this targeted analysis are written in Matlab. These codes take as input coarse or fine heterodyne narrowband time domain sequences that are generated by Glasgow-LSC group software. The codes can also take as input narrowband time domain sequences generated by the PSS Rome group software.

The codes are available at Cascina CVS repository at

`https://wwwcascina.virgo.infn.it/cgi-bin/cvsweb/cvsweb.cgi/PolgrawTargetedSearch/`

5.3.7 Direct searches, for isolated neutron stars of known positions

I) Pisa pipeline

This pipeline takes as an input h reconstructed data with quality flags. It can work using several sampling rates for input data, but for reason of efficiency should use 4 kHz data if possible. The output produced is given by a set of spectra, one for each value of the source parameters (excluding the frequency). Correction for Doppler modulation and spin down is done on a narrow (few Hertz) band around a frequency of interest using a fast resampling algorithm which can generate several resampled streams (with a low sampling rate) in parallel.

5.3.8 Searches for binary neutron stars: polynomial search

This pipeline takes as an input h reconstructed data with quality flags. The output produced are files with the parameters of the candidates.

5.3.9 Summary tables for CW

Table 5.16 gives the status of each pipeline and people involved in the year 2013, the goal for the project by Jan. 2015. Table 5.17 gives detailed milestones for the project. Table 5.18 summarizes the main computing features, for CW.

Pipeline	SVN Vers	Responsible and Collaborators	Status of the project (July 2013)	Goal (Jan. 2015)
Frequency Hough (PSS)		*P. Astone* A. Colla, S. D' Antonio S. Frasca, C. Palomba	Pipeline nearly completed Review nearly completed	Pipeline ready Review completed
Polgraw AllSky		*A. Krolak*, M. Bejger, K. Borkowski, O. Dorosh	Pipeline completed Review nearly completed	Pipeline ready to analyse data
Rome Targeted (PSS)		*C. Palomba* P. Astone, A. Colla, S. D'Antonio, S. Frasca	Pipeline completed Review completed	Pipeline ready to analyse data
Polgraw Targeted		*A. Krolak*, M. Bejger	Pipeline completed Review completed	Pipeline ready to analyse data
Direct searches		*I. Ferrante*, O. Torre, G. Cella	Pipeline under validation	Pipeline ready to analyse data
Polynomial searches		*H. J. Bulten*, R. Jonker	Pipeline under validation	Pipeline ready to analyse data

Table 5.16: Summary Table for CW pipelines.

Pipeline: milestones	October 2013	January 2014	May 2014	October 2014	Jan 2015
Frequency Hough (PSS)	Review in progress	Review completed	MDC stage 1	MDC stage 2	Ready for analysis. MDC stage 3
Polgraw AllSky	Review completed	MDC stage 1	MDC stage 2		MDC stage 3 completed
Rome Targeted (PSS)	Review completed	MDC completed on O(100) sources	Extension to narrow-band searches completed and reviewed		Pipeline ready to analyze AdV data
Polgraw Targeted	Review completed	Once pulsar spin frequency (1f) search implemented	Network search implemented	Hardware and software tests completed	Low frequency 1f and network search completed
Direct searches	Validation completed	Automatization of search over parameters	GRID version	Ready for analysis	
Polynomial searches	Validation for data with gaps	Sco X-1 MDC completed	Code review	Methods paper	Pipeline ready to analyze AdV data

Table 5.17: Milestones for CW software. MDC indicates the "Mock Data challenge" effort, carried on in the CW group using software injected signals. The goal for the different stages have been defined in the group and can be found at https://wiki.ligo.org/CW/MockDataChallenge.

Pipeline	Processors e.g. CPUs/GPUs	Submission method e.g. interactive/batch/ /GRID/Cloud/Dirac	Language/Sofware e.g. C, C++, Matlab,Root	Software dependencies CMT,Fr,FFTw
Frequency Hough (PSS)	CPUs/ GPUs	GRID	C C++	Matlab compiler
Polgraw AllSky	CPUs	batch/GRID	C, Pascal, Matlab	FFTW
Rome Targeted (PSS)	CPUs	interactive/batch/grid	Matlab	matlab compiler
Polgraw Targeted	CPUs	interactive/batch	Matlab	-
Direct searches	CPUs	interactive/batch GRID	C++, python	Fr, FFTw
Polynomial searches	CPUs	GRID/PBS	C++, python, pyROOT	FFTw, ROOT, Boost

Table 5.18: Summary Table for CW software main characteristics

5.3.10 Stochastic: Isotropic searches and Spherical Harmonics analysis

The stochastic searches share the same infrastructure. They take as input h(t) data for two detectors. Because of the reduction of sensitivity at high frequency due to the separation and the relative orientation of the detectors, the frequency range used for the LIGO Hanford/Livingston pair (HL) is 10-150 Hz, and thus the data are sampled to 1024 Hz to save computational time. Virgo/LIGO pairs performing better than HL after 200 Hz, this pair is used to probe higher frequencies and the data are sampled at 4 kHz. The software is written in Matlab as part as the Matapps package. The output files are txt or mat files. Let's notice that non reviewed but fully tested C and C++ versions of the isotropic pipeline have been developed in Virgo as part of the lalapps and NAP packages.

5.3.11 Summary tables for Stochastic

Table 5.19 gives the status of each pipeline and the people involved in the project in the year 2013 and the goal for the project by Jan. 2015. Table 5.20 gives detailed milestones for the project. Table 5.21 summarizes the main computing features.

Pipeline	SVN Vers	Responsible and Collaborators	Status of the project (July 2013)	Goal (Jan. 2015)
Isotropic pipeline	9148 but evolving	T. Regimbau	fully reviewed	Cleaned version ready to analyze data
Spherical Harmonic		T. Regimbau	fully reviewed	Cleaned version ready to analyze data

Table 5.19: Summary Table for stochastic pipelines needs.

Pipeline: milestones	October 2013	January 2014	May 2014	October 2014	Jan 2015
Isotropic pipeline	fully reviewed	cleaning			ready
Spherical Harmonic	fully reviewed				ready

Table 5.20: Milestones for stochastic software

Pipeline	Processors e.g. CPUs/GPUs	Submission method e.g. interactive/batch/ /GRID/Cloud/Dirac	Language/Sofware e.g. C, C++, Matlab,Root	Software dependencies CMT,Fr,FFTw
Isotropic pipeline	CPUs	GRID	Matlab	Fr/FFTW
Spherical Harmonic	CPUs	GRID	Matlab	Fr/FFTW

Table 5.21: Main computing features for stochastic work

5.4 More on the GWTools GPU project

GWTools is a C++/OpenCL based Gravitational Wave data analysis Tool kit. It is an algorithm library aimed to bring the immense computing power of emerging many-core architectures, such as GPUs, APUs and many-core CPUs, to the service of gravitational wave research. GWTools is a general algorithm library intended to provide modular building blocks for various application targeting the computationally challenging components of g.w. data analysis pipelines.

GWTools itself does not represent a distinct, standalon analysis pipeline. Executables built on the top of GWTools libraries can be inserted into any existing analysis flow as a drop-in replacements, enhancing the speed and extent of usability of the pipeline.

Details and status reports at http://www.gwtools.org.

5.4.1 Summary table for GWTools GPUs project

Table 5.22 gives the status of each pipeline and the people involved in the project in the year 2013 and the goal for the project by Jan. 2015. Table 5.23 gives detailed milestones for the project. Table 5.24 summarizes the main computing features.

Pipeline	SVN Vers	Responsible and Collaborators	Status of the project (July 2013)	Goal (Jan. 2015)
GWTools (CBC)	no	Gergely Debreczeni	Prototype applications are running	Fully tested ready for analysis
GWTools (CW)	no	Gergely Debreczeni, Alberto Colla, Sabrina D' Antonio	Prototype applications are running	Fully tested ready for analysis

Table 5.22: Summary Table for GWTools needs.

Pipeline: milestones	October 2013	January 2014	May 2014	October 2014	Jan 2015
GWTools for CBC	Prototype apps are running	Optimized, full performance	Fully tested, ready for analysis		
GWTools for CW	Prototype apps are running	Optimized, full performance	Fully tested, ready for analysis		

Table 5.23: Milestones for GWTools software

Pipeline	Processors e.g. CPUs/GPUs	Submission method e.g. interactive/batch/ /GRID/Cloud/Dirac	Language/Sofware e.g. C, C++, Matlab,Root	Software dependencies CMT,Fr,FFTw
GWTools for CBC	CPU and GPU	Cluster and Grid	OpenCL,C++	lalsuite
GWTools for CW	CPU and GPU	Cluster and Grid	OpenCL,C++	none

Table 5.24: Main computing features for GWTools work

5.5 Software to store IGWD data: the frame files

The format used to store the data is a collection of “frames”, where the time duration depends on the channel to be stored. The format is common to LIGO. It is described in [2] and in [6]. Frames are written assuming IEEE/ASCII compliant hardware and software. This standard specifies the organization and content of “Interferometric Gravitational Wave Detectors” (IGWD) Frame data sets, including the C structures which define a frame. LIGO and VIRGO have agreed to work to ensure that all developed hardware and software systems will support IGWD Frames for the interchange of binary data. All participating projects will acquire their data in Frames and make their data available, when and if data exchanges occur, in Frame formatted files. Reduced data still containing time-series representation of IGWD datastreams shall be made available in Frames.

5.6 AdV official software

5.6.1 Management

To ensure quality and maintenance, data analysis software developments must follow some rules inherited from good software engineering practices. Software management tools have also been adopted long time ago in Virgo. Similar but adapted rules and tools should apply to data analysis software developments:

- Package organization: software (written in any high level language code or interpretated language scripts) must be organized in packages that gathers source, documentation and compiling/installing instructions. Stricter rules about package organization are edicted by the package configuration tools chosen by Virgo (see below). A package is identified by a version. The Virgo convention is v<X>r<Y> or v<X>r<Y>p<Z>. If the changes in the library of the package are not backward compatible, the "version" number must be increased by one unit. Any backward compatible changes imply an increase of the "release" number. Package versions are listed in Tab. 5.25.

- Configuration Management tool: Virgo has adopted CMT to manage its software. This tool allows to identify the configuration of a package and its environment (operating system, ...). It provides tools to control systematically changes and dependences between packages and finally compile and install Virgo software in a easy way. CMT instructions are defined in each package in a requirements files contained in a /cmt folder. Not every Virgo software is managed by CMT (see Tab. 5.25) but this should be uniformed in the near future.

- Revision control tool: for years, Virgo used CVS to control software versions and developments. Recently a migration to SVN has been performed. In addition to commit and add comments for each file change, it is recommended to keep a summary of the main changes in a ChangeLog file stored at the root of the package. The Virgo SVN repository is hosted in a server at Cascina.

- Change and bug report system: Virgo has adopted SPR (Software Problem Report). This tool has been adapted for Virgo. It's meant to raise attention and notify the relevant person concerned by an issue that affect anyone else. It also helps the Virgo software manager to keep track of bugs and corrections. This tool is mainly used for DAQ/control software but should also be used for data analysis software, although bugs report and fixing are usually following another cycle.

package name	version (Sept. 2013)	SVN (Sept. 2013)	CMT (Sept. 2013)
DMS	-	YES	YES
Spectrograms	-	NO	NO
MonitoringWeb	-	YES	NO
Omicron	v1r1	YES	YES
NoEMi	-	YES	NO
NMAPI	-	YES	NO
UPV	v2r3	YES	YES
Excavator	-	NO	NO
DQPerf	v1r1	YES	YES
cWB	-	LIGO repository	NO
STAMP-AS	-	LIGO repository	NO
X-pipeline	-	LIGO repository	NO
MBTA	v1r71	YES	YES
iHope	-	LIGO repository	NO
PolgrawAllSky	-	YES	NO
PSS	-	YES	YES
TIGER	-	?	NO

Table 5.25: Status of AdV software.

5.6.2 Data analysis software development and release

Many data analysis software are developed outside Cascina (laptop or computing centers). The rules given above apply to any Virgo software. The situation is more complicated when the software is jointly developed with LSC members (for instance the LSC is using autoconf for package configuration).

So far, a large fraction of the data analysis software is not managed directly by the Virgo software manager, because data analysis groups have different practises (internal review and especially code review, use of the LSC archiving tools, ...). This is less the case of software running at Cascina, but the current data analysis release does contain only a fraction of Virgo data analysis packages developed by Virgo users. adV CCs situation is currently different: CNAF is directly managed by the Virgo software manager, while CCIN2P3 situation is more development oriented (no tagged release installed, but CVS installation is done by developers in a common disk area).

5.6.3 Interoperability with LIGO software

As already mentioned, Virgo and LSC software are managed differently and both management are not compatible. One of the difficulties holds in the package configuration. The LSC has chosen the autotools (autoconf and automake) open source software. Another source of diffculties comes from that many LSC pipelines are strongly dependent of the batch scheduler system (*condor*) used on all LSC clusters. condor is not installed in any Virgo computing centers. This is one of the reasons why pipelines that have been initially developed by the LSC are run almost all on LSC clusters and not Virgo Tiers 1.

5.6.4 Operating systems

AdV has made the choice to support software on Scientific Linux. That also corresponds to CNAF and CCIN2P3 cluster operating systems, following CERN choice. Currently SL5 is installed on Cascina servers. Because of the end of maintenance service and because of the storage farm middleware upgrade, a migration to SL6 is needed and already planned (2014), but at the horizon of

2016 (Virgo upgrade to dual recycling breakdown), SL7 will be deployed on all Cascina servers. At CCIN2P3, the migration to SL6 is not foreseen before 2013-2014.

5.7 Data transfer (DT) software

The technical details on the procedures will be described in the Implementation Plan, while the requirements have been detailed in Part III of this CM.

5.7.1 Summary tables for Data transfer

Table 5.26 gives the status of each pipeline and the people involved in the project in the year 2013 and the goal for the project by Jan. 2015. Table 5.27 gives detailed milestones for the project. Table 5.28 summarizes the main computing features.

Pipeline	SVN Vers Vers.	Responsible and Collaborators	Status of the project (July 2013)	Goal (Jan. 2015)
Low-latency				
Bulk		EGO	basic tools to CC1,CC2 ready; web monitoring work in progress. h(t) copy to CC* to be defined	tool ready for production
aLIGO to AdV		EGO or Virgo, depending on the strategy	strategy to be defined	tool ready for production
AdV to aLIGO		EGO	ongoing work with LIGO colleagues	tool ready for production

Table 5.26: Summary Table for Data transfer tools.

Pipeline: milestones	October 2013	January 2014	May 2014	October 2014	Jan 2015
Low-latency					
Bulk DT	web monitor work in progress	web monitor completed	file catalog integrated	stress test	tool ready
aLIGO to AdV	studies on possible strategies	tests on two strategies	stress test		
AdV to aLIGO	discussions with LIGO	test the tool	stress test		

Table 5.27: Milestones for data transfer software. Regarding the DT from aLIGO to AdV we have two possible scenarioes: aLIGO to Cascina, a solution which already works, under the EGO responsability; aLIGO to AdV CCs, a solution to be exploited by Virgo members.

Pipeline	Processors e.g. CPUs/GPUs	Submission method e.g. interactive/batch/ /GRID/Cloud/Dirac	Language/Sofware e.g. C, C++, Matlab,Root	Software dependencies CMT,Fr,FFTw
Low-latency	CPU			
Bulk	CPU	batch	C, Python	lcg-tools, irods
aLIGO to AdV	CPU	batch	C, Python	tbd
AdV to aLIGO	CPU	batch		LDR tools

Table 5.28: Main computing features for Data Transfer work. LDR indicates the “LIGO data Replicator” tool, http://www.lsc-group.phys.uwm.edu/LDR/

5.8 Data management (local and remote access) software

5.8.1 Summary tables for Data management work

Table 5.29 gives information on the data access resources and milestones.

The Data Management project will be unitary and will guarantee a transparent access to the data, but here we have divided the work into its two main sub-parts, as they will have different milestones. One regards the access to the data by the scientific pipelines, where the data to be accessed are not a huge quantity, O(10) TB, there is no need for graphical monitoring tools and the access to the data is through tools like GRID/CLOUD, that is not interactive.

The second regards all the other activities, e.g. commissioning, detector characterization, where graphics tools are very important and the data accessed are typically huge (O(100) TB).

Pipeline	SVN Vers Vers.	Responsible and Collaborators	Status of the project (July 2013)	Goal (Jan. 2015)
Scientific pipelines	TBD	TBD	TBD	TBD
Others, e.g. commissioning, detchar	TBD	TBD	TBD	TBD

Table 5.29: Summary Table for Data management tools, having divided the item into two main sub-parts, as explained in the text, to be able to describe detailed milestones for each of them. TBD stands for “to be defined”.

Chapter 6

Databases

6.1 Databases description

Virgo uses mysql databases...etc...etc...

6.2 Detector Monitor System

6.2.1 Description

The DMS is a tool based on a set of monitoring processes providing the needed information and a WEB tool intended to collect, further process, archive and display/notify, in a centralized way, the information regarding the status of the ITF, the technical infrastructures, the electronics equipment, etc... This information is displayed as a 'multi cell' table, where each cell, named flag, denotes the status (codified by a color) of a specific item. In case of problem the system can perform several types of notification.

Summary		
ADMIN	DB Name:	DetMoni2
	Releted Project:	Detector Monitor System
	Reference person:	F. Berni
DBMS	DBMS Required:	MySQL
	DBMS Version Required:	5+
ACCESS	List any required DB-level users and their associated permissions:	
	Will the DB be made available only within the Virgo (Cascina) network, or also externally?	Cascina + External
USAGE	What type of usage will be made of the DB:	Intensively used
SUPPORT	What kind of back-up strategy should be applied to the DB?	Daily
	Does the DB require on-call support?:	Yes, it is critical and in the event of failure it must be brought back online as soon as possible.
FORECAST	How much required disk space is foreseen for the DB over the following periods after the application is put in production?	
	How many daily requests to the DB are foreseen over the following periods after the application is put in production?	
	How much, on average, CPU is foreseen in the handling of requests to the DB over the following periods after the application is put in production?	

6.3 Condor Job Monitor Interface

6.3.1 Description

This DB stores information about the status of the submitted Condor Jobs.

Summary		
ADMIN	DB Name:	CondorJob
	Releted Project:	Job Condor Monitoring Interface
	Reference person:	F. Berni, G. Hemming
DBMS	DBMS Required:	MySQL
	DBMS Version Required:	5+
ACCESS	List any required DB-level users and their associated permissions:	
	Will the DB be made available only within the Virgo (Cascina) network, or also externally?	Cascina + External
USAGE	What type of usage will be made of the DB:	Intensively used
SUPPORT	What kind of back-up strategy should be applied to the DB?	Daily
	Does the DB require on-call support?:	NO
FORECAST	How much required disk space is foreseen for the DB over the following periods after the application is put in production?	
	How many daily requests to the DB are foreseen over the following periods after the application is put in production?	
	How much, on average, CPU is foreseen in the handling of requests to the DB over the following periods after the application is put in production?	

6.4 Lines

6.4.1 Description

Backup of the old Lines metadata database.

Summary		
ADMIN	DB Name:	lines
	Releted Project:	Lines
	Reference person:	G. Hemming, I. Fiori
DBMS	DBMS Required:	MySQL
	DBMS Version Required:	5+
ACCESS	List any required DB-level users and their associated permissions:	
	Will the DB be made available only within the Virgo (Cascina) network, or also externally?	Cascina
USAGE	What type of usage will be made of the DB:	Never used
SUPPORT	What kind of back-up strategy should be applied to the DB?	Monthly
	Does the DB require on-call support?:	NO
FORECAST	How much required disk space is foreseen for the DB over the following periods after the application is put in production?	
	How many daily requests to the DB are foreseen over the following periods after the application is put in production?	
	How much, on average, CPU is foreseen in the handling of requests to the DB over the following periods after the application is put in production?	

6.5 Data Quality Segment Database

6.5.1 Description DQSEGDB

Stores DQ segments that are produced either online or offline. These segments are time periods defining either the global status of the interferometer (science, not locked, etc.) or containing transient statements about the operation of the instrument or quality of the data that may impact upon data analyses.

Summary		
ADMIN	DB Name:	
	Releted Project:	Data Quality Segment Database (DQSEGDB)
	Reference person:	G. Hemming (D. Verkindt and M.-A. Bizouard may also be contacted)
DBMS	DBMS Required:	Not yet defined. Current version is in MySQL, but the server writing to and reading from the database is using ODBC, as PostGreSQL or another DBMS may be used.
	DBMS Version Required:	If MySQL, at least 5. Latest stable release preferable. If PostGreSQL, still to be determined. But likely to be latest stable release.
ACCESS	List any required DB-level users and their associated permissions:	dqsegdb_user@localhost, dqsegdb_readonly@localhost, Others TDB
	Will the DB be made available only within the Virgo (Cascina) network, or also externally?	The database will be intensively used, with all access to it from command line or web interface being funneled through a Python server.
USAGE	What type of usage will be made of the DB:	Daily
SUPPORT	What kind of back-up strategy should be applied to the DB?	Daily
	Does the DB require on-call support?:	YES
FORECAST	How much required disk space is foreseen for the DB over the following periods after the application is put in production?	
	How many daily requests to the DB are foreseen over the following periods after the application is put in production?	
	How much, on average, CPU is foreseen in the handling of requests to the DB over the following periods after the application is put in production?	

6.5.2 Description DQSEGDB WUI

Stores textual and metadata content relating to the web interface to the DQSEGDB.

Summary		
ADMIN	DB Name:	dqsegdb_wui
	Releted Project:	Data Quality Segment Database (DQSEGDB)
	Reference person:	G. Hemming
DBMS	DBMS Required:	MySQL
	DBMS Version Required:	5+
ACCESS	List any required DB-level users and their associated permissions:	dqsegdb_wui@localhost
	Will the DB be made available only within the Virgo (Cascina) network, or also externally?	TBD. Currently only available internally. Extent to outside availability still to be discussed in depth.
USAGE	What type of usage will be made of the DB:	The database will be intensively used during run time, but also used out of run as well.
SUPPORT	What kind of back-up strategy should be applied to the DB?	Daily
	Does the DB require on-call support?:	TBD
FORECAST	How much required disk space is foreseen for the DB over the following periods after the application is put in production?	
	How many daily requests to the DB are foreseen over the following periods after the application is put in production?	
	How much, on average, CPU is foreseen in the handling of requests to the DB over the following periods after the application is put in production?	

6.6 NOEMI

Database	tables	records	Size
Events-<Run >	5*n. of channels	1 per day (NOTE)	few KB/day
Lines	7	O(1E6)/year	O(100 MB)/year

NOTE: Currently there is 1 record per peak in the peak-map but this is considered unmanageable. We will re-organise the database, writing the peakmaps data on disk and storing in the DB only the pointers to the files (1 per day).

6.7 WDF

Database	tables	records	Size
WDF	12	14.809.530	794.5 MB

6.8 COHERENCES

Database	tables	records	Size
Coherences	4110	4.167.558.999	121.6 GB

Chapter 7

User credentials

7.1 User credentials

People in the Virgo Collaboration need to access resources ranging from ssh services, web sites and GRID User Interfaces that are spread in many administrative domains, like laboratories and computing centers, using a wide variety of credentials and authentication methods. The management of ever growing multiple access credentials for a single user and the need of authenticating to different applications in the same work session is an effort that makes more difficult the science activity, both for the end user but also for the AAI (Authentication Authorization and Identity) infrastructure administrators. This problem has been therefore fronted by many organizations with the aim of decreasing the number of credentials needed by each user while adopting SSO (Single Sign On) AAI infrastructures for the transparent authentication to the highest possible number of services.

For the Virgo organization EGO manages the identities of the Virgo users in Cascina and also hosts the Web applications to be accessed both by Virgo and LSC users. Therefore it has started a revision of the AAI system that will allow the use of the users's home institutions identities for the Web access using the standard "SAML" protocol and the identity federations based on it. It is also the natural entity to manage the Virgo "Virtual Organization" (in the identity federation sense), centralizing the administration of the Virgo users attributes. Among the services that could benefit from the use of the Virgo users federated identity there will be also the GRID access, whereas it can be mediated by a generic web portal (such as the one in development by IGI, Italian Grid Initiative) or by a yet-to-develop "Science Gateway" dedicated to AdVirgo GRID applications. The path to this final scenario is not straightforward, in that the identity federations involved are multiple, and spread in various countries (IDEM for INFN and the other Italian groups, FER for CNRS and the other French groups, Ligo.org/InCommon for LSC, etc.) and there is not yet in place an infrastructure that covers the collaborations across these boundaries (although the EduGAIN inter-federation is reaching the majority of European countries).

For this reason EGO, once completed the upgrade of the internal IdM (Identity Management system, will collaborate with LSC to find shortcomings solutions for the mutual federation that don't impact on the main scenario.

Unfortunately no solutions are foreseen for the problem of the direct interactive access to the computing resources, or user interfaces, in use by Virgo around the world. These computing resources fall inside different administrations, each one requiring its own account issuing process.

Part V

Computing facilities resource requirements

Chapter 8

Cascina and AdV CCs

8.1 Computing farm for commissioning and analysis

We summarize here the storage needs in Cascina and describe the computing needs for the online, in-time detector characterization activities and the science low-latency searches.

8.1.1 Storage needs at EGO/Cascina: summary tables

We refer here to the Data Model described in Section 2.2, Section 2.3, Section 3.1.2.2, to specify in Table 8.1 the storage requirements in Cascina.

Data	Buffer length years	Storage in Cascina for 1 year [TB]
Full Bandwidth raw	0.008	12
Raw data	0.5	385
AdV RDS	1	11
Trend data	3	5
Minute trend data	3	1
AdV h(t) and status flags	1	3
LIGO h(t) and status flags	0.5	3
Calibration output	permanent	1 TB (+10GB/yr)
DS	space limited	30
Omicron triggers	1	2
Online veto production data	permanent	negligible
Spectrogram data	1	1
MonitoringWeb data	permanent	0.8
DQ developments data	–	0.5
DQ segment	–	negligible
NoEMi data	1	12
SILeNTe data	negligible	negligible
Regression data	0.1	0.4
WDF data	1	0.1
NonStatMoni data	1	0.2
Coherence data	1	0.2
MBTA output data	1	0.5
Total	—	488.7

Table 8.1: Summary Table: storage needed in Cascina (IGWD data, detchar data, low-latency searches)

The total needed storage, considering one year of data taking and a duty cycle of 100%, is thus 488.7 TB. As shown in table 8.7 these data are copied to the AdV CCs, with the rules defined in Part III of this computing model. From the gained experience during Virgo, we know that the commissioning team needs to have on-site at least 6 months of recent data to quickly investigate the ITF behaviour, using monitoring tools running in Cascina. We would notice that the power of the farm needed in Cascina is not affected by this choice, as it is dominated by the detchar on-line and in-time analyses.

8.1.1.1 Comments on the storage needs for Commissioning and calibration data

8.1.1.2 Comments on the storage needs for Detector characterization

- Omicron will need, for 1 year of science run, about 2 TB to store the triggers of 600 channels at Cascina. A local storage over 2 years (thus 4 TB) is required to deal with data quality follow-up around some events output by off-line analysis. In parallel, the Omicron triggers will be transfered to the Lyon CC for permanent archive.
- MonitoringWeb (including spectrograms) will need, for 1 year of commissioning or science run, about 500 GB to archive the various plots daily. It is expected to keep those archive at least over 2 years.
- Spectra data produced by SpectroMoni require about 1 TB to store 1 year of commissioning or science run. For any data quality follow-up or spectrogram reprocessing, those data should be kept at Cascina over at least 2 years. In parallel, the spectra data will be transfered to the Lyon CC for permanent archive.
- Other detector characterization data like Omiscans, UPV and Excavator results, DQ performances, Omicron web pages, DQ segments stored in DQSEGDB, DMS archives, DQ developments and tests, require a total of a few hundred of GB each year. For most of those data, it may be useful to keep them in Cascina for at least 2 years. A priori, there is no need to transfer and archive all these data in an external CC (e.g. IN2P3). The data created in Cascina and exported to CCs have been indicated in the Data Model part of this Computing Model.

8.1.2 Computing needs at EGO/Cascina

The architecture implementation in Cascina should allow to share the data present in shared memories among most of the processes, to avoid the need to transfer the data between machines and processes. The online and the offline machines in Cascina should have the same architecture. They will be put in the same farm of a few large-CPU/large-RAM machines with virtualization to ease the resource management. However, the critical online data collection processes must run on one or two machines separated from the other as a separated hardware. We need to have machines dedicated to the control room, one or two machines with large CPU and RAM and which guarantee the possibility to start different sessions with different screens and keyboards.

Such a configuration would allow to have the raw data available online in shared memories directly on the machine for a faster visualization in the control room.

To run the online and in-time detector characterization analysis we need to have dedicated machines, for some, and a batch system, for others. In the following, one core means a typical one currently (2012/2013) used in Virgo: Opteron 275 at 2.2GHz associated to at least 1 GB of RAM. We have used the conversion 1 core = 10 HS06 and to get the energy integrated over 1 year we have used the following equation: energy=$365 \times N_c \times \frac{T}{24} \times 10$ HS06.day, where N_c is the number of cores; T the number of hours during one day when the pipeline is active (as usual, having considered the detector on with 100% duty cycle).

8.1.2.1 Computing needs for Commissioning and calibration

8.1.2.2 Computing needs for Detector characterization: Data Quality

One of the main computing resources will be to run the Omicron pipeline online over about 600 channels. This will require about 60 cores full time. Additional computing will be needed, for a total of about 40 cores, subdivided into DMS (4 cores), dataDisplay server (4 cores), MonitoringWeb (4 cores), Spectrograms (16 cores), DQ production, monitoring and performance estimation (4 cores), Omiscans and UPV and Excavator (2 cores), DQ developments (6 cores).

Table 8.2 summarizes the needs.

8.1.2.3 Computing needs for Detector characterization: Noise studies

Table 8.3 summarizes the needs.

8.2 Tables of CPU needs at EGO

8.2.1 Detector characterization: Data quality

Analysis	Cores number	Time, in hours/day	kHS06.day integrated over 1 yr	Power kHS06
Omicron pipeline	60	24	219	0.6
On-line vetoes	4	24	14.6	0.04
DMS	4	24	14.6	0.04
Spectrograms	16	20	48.6	0.13
MonitoringWeb + DQ monitoring	6	12	10.9	0.03
dataDisplay server	4	2	1.2	0.003
DQ developments studies	6	0.01	negl.	negl.
Total	100	–	309	0.84

Table 8.2:

8.2.2 Detector characterization: Noise studies

Analysis	Cores number	Time, in hours/day	kHS06.day integrated over 1 yr	Power kHS06
NoEMi	100	3	45.6	0.125
SILeNTe	50	12	91.3	0.25
Regression	50	3	22.6	0.06
WDF	50	24	182.5	0.5
NonStatMoni	10	24	36.5	0.1
Coherence	50	3	22.6	0.06
Total	-	-	401.1	1.1

Table 8.3:

8.2.3 Low Latency Science analysis

There is only one low-latency search which will run in Cascina, the MBTA pipeline, in Section 1.4. The Computing needs to run this pipeline are summarized in Table 8.4.

Analysis	Cores number	Time, in hours/day	kHS06.day integrated over 1 yr	Power kHS06
MBTA	32	24	116	0.32
Total	32	24	116	0.32

Table 8.4:

8.3 Summary table of CPU needs at EGO

Analysis	Core number	kHS06.day integrated over 1 yr	Power kHS06
Commissioning: Calibration			
Detchar: DQ	100	336	0.92
Detchar: Noise	110	401.5	1.1
Science low-latency analysis			0.32
Total			

Table 8.5: Summary table of CPU needs at EGO/Cascina

We describe here the storage and computing needs in AdV CCs. The numbers refer to a run of 1 year. We need to have copies of all data which cannot be reproduced again or which cannot be easily reproduced (meaning intensive CPU usage, intensive human activity) in at least two CCs. The raw, RDS and h(t) data are copied in two CCs also for redundancy reasons: these data are stored in the Tier-0 only for a period of 6 months and are not backupped there (as said, there is a crash recovery backup to cover the period of time before the data transfer to CCs).

We foresee to continue to work in the CCs as done for years with the Virgo detector, with tapes and cache disks. In the table we have not specified what will be needed on disk and on tape but from the experience in Virgo we foresee:

- to store all the commissioning, science and astrowatch data permanently on tape. This implies to yearly increase the storage on tape by an amount which will depend on the run time of the detector;
- To have on disk all the data taken in the last run of the detector. At regime, when the detector will take data continuously for 1 year this will mean to have a disk storage of 1 PB.

The paper from the two LIGO and Virgo collaborations, at http://arxiv.org/abs/1304.0670, gives our best estimation on how the run durations will evolve during the first years of Advanced

Detectors Era.
These requirements will be updated every year.

8.4 Storage needs in AdV CCs

Table [8.6] gives the summary of the requirements to store *one replica* of the data (first column) and the total storage needed following the guidelines stated before (second column), at regime situation (1 yr of data taking).

Data	one replica TB	Total TB
Raw data	745	1490
AdV RDS	11	33
aLIGO RDS	22	66
Trend data	1.5	4.5
Minute trend data	0.025	0.075
AdV h(t) and status flags	3	9
aLIGO h(t) and status flags	6	18
MDC h(t)	12	36
Calibration output	1	3
Omicron triggers	2	2
DQ veto production data	negligible	negligible
Spectrogram data	1	1
MonitoringWeb data	0.8	0.8
DQ developments data	0.5	0.5
DQ segment	negligible	negligible
NoEMi data	12	24
BURST	18	36
CBC	4.5	9.0
CW	25	50
STOCHASTIC	3.6	3.6
Total	868.92	1786.5

Table 8.6: Summary Table: storage needed, for only one replica of the data and storage needed for multiple copies, We have considered to copy raw data and other data which cannot be easily reproduced in a minimum of two CCs and to copy of h(t) and the analysis intermediate results in three CCs, which is the minimum we foresee to have.

8.4.1 Storage needs in AdV CCs, from 2015 up to 2017

We need to re-scale the numbers in the above table, considering the expected run time and also considering the ongoing analyses on Virgo data. To summarize:

to store one copy of 1 yr of LIGO data only we need 28 TB (as only h(t) and RDS are copied in our CCs);

to store one copy of 1 yr of all the Virgo and LIGO data we need (800-850) TB;

to store one copy of 1 yr of Virgo data needed for CPU demanding pipelines we need 105 TB.

The request of storage on tapes is incremental, that is every new run needs to be stored on tape for ever, while the request of disk space is not integrated over time.

Hence:

- 2014:

  No new data are taken. Considering that the Virgo data are already stored in two CCs we would need to have 105 TB additional disk storage in a third CC, where to run under GRID the ongoing CW and CBC analyses.

- 2015:

  Given that LIGO will run a 3 months data taking and that Virgo will have some commissioning data we have estimated an additional need of the order of 20 TB in each CC (having considered the 2014 request of 105 TB in a third CC fulfilled) .

- 2016: LIGO and Virgo will take scientific data for 6 months, but we will have 6 months of commissioning data,

  which means the need for 800 TB in one CC (the one mainly used for data quality work) , 850 in a second CC (the one mainly used under GRID/CLOUD, with limited local access) and only 53 TB in the third CC, mainly used for highly demanding pipelines, with local access, GRID/CLOUD and LSC complaint tools. The 800 and 850 TB storage will need to be on disk and also on tape, for future access and backup reasons.

- 2017: LIGO and Virgo will take scientific data for 9 months, but we will have 3 months of commissioning data, which means the need for 800 TB in one CC (the one mainly used for data quality work) , 850 in a second CC (the one mainly used under GRID/CLOUD, with limited local access) and only 80 TB in a third CC, mainly used for highly demanding pipelines, with local access, GRID/CLOUD and LSC complaint tools. The 800 and 850 TB storage will need to be on disk and also on tape, for future access and backup reasons.

Table 8.7 gives some details, that is the summary of the requirements to store data in the current CCs, CNAF and CCIN2P3 (having in mind the actual searches carried on in the two CCs)

Data	CNAF (CC2) [TB]	CCIN2P3 (CC1) [TB]
Raw data	745	745
AdV RDS	11	11
LIGO RDS	22	22
Trend data	1.5	1.5
Minute trend data	0.25	0.25
AdV h(t) and status flags	3	3
MDC h(t)	9	9
Calibration output	1	1
Omicron triggers	–	4
DQ veto production data	–	negligible
Spectrogram data	–	1
MonitoringWeb data	–	0.8
DQ developments data	–	0.5
DQ segment	negligible	negligible
NoEMi data	12	–
BURST	16	3
CBC	4.5	0.5
CW	25	–
STOCHASTIC	–	3.6
Total	850.5	802.8

Table 8.7: Summary Table: storage needed in the CCs

8.5 Computing needs for off-line analyses

Given the fact that, as will be clarified in the next tables, the CPU requests is going to become larger and larger, to cover increasingly needs for the parameter space of these searches, we strongly support the possibility to use more computing centers, in particular aiming to add computing resources under GRID/CLOUD.

Another very important need is to increase the availability and usability of the computing resources and enable the usage of a Grid-enabled, LIGO-compatible Condor cluster for Virgo people, having also local submission capability.

Of course, the various computing and storage tasks have to be coordinatedly distributed over these CCs, as detailed in Section 3.1.

All the following numbers refer to the analysis of 1 year of data, taken with Duty Cycle of 100%. The requests will be yearly refined, as we have done so far for the Virgo detector.

8.5.1 Detector characterization: Data quality

The reprocessing of the online DQ flags will be done mainly in the Lyon CC. This is not supposed to be CPU expensive but the faster we get the result, the faster we provide checked reprocessed DQ flags to the data analysis groups.

A few DQ flags, like HREC or SCIENCE will be reprocessed using the trend data or raw data available on the Cascina site.

The reprocessing of the Omicron triggers, if needed, will be done in the Lyon CC. In this case, we will need to reprocess X months of data in less than X/10 months so that the result can be available in a valuable time.

Any reprocessing of the spectro data or the spectrogram plots will be done, whenever possible, on the Cascina site. If really needed, for data older than 6 months and only for spectro data reprocessing, we may use the raw data archived at Lyon CC. The needed CPU will be quite low.

Same comment for what concern any reprocessing of MonitoringWeb plots.

Most of DQ developments will be done using the raw data available on the Cascina site but some estimation of DQ performance over a large data set (like a full Virgo scientific run) may be needed in some cases and will be done in the Lyon CC, using the raw data archived there.

Analysis	Cores number	Time, in hours/day	kHS06.day integrated over 1 yr	Power kHS06
Omicron reprocessing pipeline	600	1	91.2	25
On-line vetoes reprocessing	200	1	30.4	8.3
Spectrograms reprocessing	160	0.01	0.24	negl.
MonitoringWeb reprocessing	60	0.01	0.09	negl.
DQ developments studies	200	0.01	0.3	0.08
Total	1220	–	122.2	33.4

Table 8.8:

8.5.2 Detector characterization: Noise studies

Offline noise studies work will be performed in case of reprocessing of h(t) data or if new auxiliary channels need to be analyzed. Since it is not possible to foresee the CPU needs in this case, in Table 8.9 we put the numbers expected at EGO farm (Table 8.3).

Offline noise pipelines will run on a local batch system, or even under the GRID.

Analysis	Cores number	Time, in hours/day	kHS06.day integrated over 1 yr	Power kHS06
NoEMi	$\leq$100	$\leq$3	$\leq$45.6	$\leq$0.125
SILeNTe	$\leq$50	$\leq$12	$\leq$91.3	$\leq$0.25
Regression	$\leq$50	$\leq$3	$\leq$22.6	$\leq$0.06
WDF	$\leq$50	$\leq$24	$\leq$182.5	$\leq$0.5
NonStatMoni	$\leq$10	$\leq$24	$\leq$36.5	$\leq$0.1
Coherence	$\leq$50	$\leq$3	$\leq$22.6	$\leq$0.06
Total	-	-	$\leq$401.1	$\leq$1.1

Table 8.9:

8.5.3 Computing needs for offline scientific analysis

While the needs for detector characterization are expected to be constant, once the detector will be running some of the scientific pipelines will be highly computationally demanding and they will have varying CPU requirements during the first year or two of the data taking due to the need to accumulate data (in the case of CW searches) and to the different parameter space which can be analyzed, depending on how sensitive the detector is, how many detectors are running, and how many triggers are to be followed up (in the case of CBC analyses and Burst analyses). Given the fact that the computing requests in some cases will have an important impact on our requests to the CCs we are now detailing what will be needed in the next years and we are clarifying the underlying conditions. It must be clear that:

- the requests will be refined every year as we have done so far. Every year we will write a document with the computing and storage needs for the next year;
- the most demanding pipelines run or will run by advanced detector era under a distributed environment which will be the evolution of GRID (what actually is EGI) by the year 2015+. We would be ready to make use of some remote submission methods which will make it possible to submit jobs under GRID or CLOUD (e.g. using tools like DIRAC, see http://diracgrid.org);
- for this reason, the requests of computing power are labeled as "under GRID", without specifying which CC is involved. Details of the possibilities we are exploiting are given in the Implementation Plan.

Analysis (at regime 2018+)	Cores number	kHS06.day integrated over 1 yr	Power kHS06	Comments
cWB offline BURST	300	1100	3.0	
cWB pre-conditioning BURST	30	11	0.03	
STAMP BURST				
X-pipeline BURST				
ihope, with GWTOOLS CBC	300	1100	3.0	
LALInference Parameter estimation (PE) and TIGER CBC	3000	11000	30	to follow O(10) triggers (from 1 yr of data)
Frequency Hough (PSS) CW	3000	11000	30	τ_{min} down to 100 yrs limiting the Sky Volume
Polgraw All-Sky CW	3000	11000	30	
Rome targeted (PSS) CW	negl.	negl.	negl.	
Polgraw Targeted CW	negl.	negl.	negl.	
Direct searches CW	300	1100	3.0	
Polynomial searches CW				
Isotropic analysis STOCH	negl.	negl.	negl.	
Spherical Harmonic STOCH				
Total				

Table 8.10: Estimation of computing needs by the year 2018 +

In the next subsections we go into details to explain where the needs for high computing power arise, concentrating only on the highly demanding work.

8.5.3.1 Details on the Computing needs for the cWB All-Sky searches

As of today we plan to run the main cWB All-sky search on LSC clusters, but we would also be able to run it under our CCs. This will be possible if we shall have the manpower for the porting of the pipeline under an Architecture complaint with our CC (GRID). We plan to run under one of the AdV CCs the pre-conditioning step, for which the computational burden is not high but which needs to have a fast access to the raw data files (or, to the RDS data, if these will include all the channels needed for this analysis).

8.5.3.2 Details on the Computing needs for the Frequency Hough PSS analysis

All-Sky CW analysis are computationally bounded. For this reason the search method is always hierarchical and the codes organized in such a way that the parameter space can be reduced to fit the computing power. The requests in the table fit with a good scientific goal, which implies to exploit a parameter space which is wide enough to carry on a sensible search. As explained in [8], the age of the neutron star, the parameter τ_{min}, enters in the computing burden as $1/\tau_{min}^2$ (neglecting the effect of the second order spin-down which at small values of τ_{min} enters in the game again increasing the parameter space), which means that if 300 cores/year are enough to exploit the full sky for a value τ_{min}=10000 yrs (these are numbers obtained by running a real search at CC2, CC*,[9]) we would need 3×10^4 cores/yr to go down to τ_{min}=1000 yrs. For this reason, we have planned to run searches on τ_{min} of the order of O(100) yr, reducing the sky volume where to look for. Another possibility on which we are working is the porting of the pipeline under GPUs (GWTOOLS for CW project).

8.5.3.3 Details on the Computing needs for the Polgraw All-Sky analysis

For this search the same considerations of the previous paragraph apply. The proposed search saves computing power by reducing the frequency band to be exploited. To perform a search from the lowest available frequency upto a kHz and assuming the minimum age $\tau_{min} = 1000$yr we need 2000 cores/yr to analyse 1yr of data [11]. If we wanted to go to 2kHz we would need 3×10^4 cores/yr to analyse 1yr of data.

8.5.3.4 Details on the Computing needs for ihope with GWTOOLS

The CBC analysis using GWTools runs on CPU and GPU devices, as well. In order to exploit the computing power of GPUS and provide the collaboration with a testing / developing environment we propose the setup (or the extension of an already existing) GPU cluster. A configuration of 16 GPU card in 8 host machine (with a nominal processing power of c.c. 60 TeraFlop) would do for this purpose in 2014 and extended later on as needed.

8.5.3.5 Details on the Computing needs for LaLInference work

This is a pipeline which runs on the triggers found with the main CBC search (based on ihope). The computing burden here comes from the need to estimate with high precision the background around each trigger. Details are given in [10].

Assume that we go with the idea of running an independent background for every source detected, some time before the detection (indicated as t_c) we generate the background relative to that stretch of noise.

To have N independent catalogs of S sources (doing n_t tests (n_t is the number of testing parameters)), we need

$R = (S \times N) \times 2^{n_t}$ runs. With a run time t per run, we need

$H = R \times t$ hours of cluster. Given a certain number of cores C, we need a cluster time

$T_c = \frac{H}{C}$

Assuming that the S sources are observed over T_{obs}, we need a fraction of the total time of the cluster:

$f = \frac{T_c}{T_{obs}}$

Now, using: $n_t = 4$, N = 1000, S = 10 (2018+) in 2 years (which is quite optimistic, given the noise curves), R = 40 hr, C = 1000, $T_{obs} = 2$ yr, we get a total of H = 6400 h = 8.9 months.

Over a period of 2 years implies that we will only use those nodes f = 0.37 of their full potentiality.

Changing N to 2500, we get f = 0.92. The same efficiency is obtained also for n_t=3 and N = 5000. To summarize: 2 years of running on 1000 nodes for 37% of the time will allow analysis of 9 months' worth of data. For this reason we have estimated that with 3000 cores we can analyze O(10) triggers in the much reasonable time of 2/3 year (8 months, comparable to the supposed observing time).

8.6 Computing needs: summary tables in regime situation (2018+) in AdV CCs

Pipeline needs in kHS06 power	local	GRID/CLOUD
Detchar Data Quality	1	–
Detchar Noise studies	1	1 ?
BURST	negl	3
CBC	–	33+
CW	–	60+
STOCHASTIC	negl.	negl.
TOTAL	2 ?	97+

Table 8.11: Summary Table: Estimation of the computing needed locally in the CCs and under GRID/CLOUD at a regime situation (2018+), under certain hypotheses on the parameter space covered. Units are power in kHS06. The "+" indicates that this is the minimal request, with more resources we could cover a wider parameter space

8.7 Estimation of yearly computing and storage needs in AdV CCs from 2014 to 2017

In these years some of the computing power will be needed to complete the analysis of the Virgo data and to do tests in preparation of the full sensitivity of the detectors in ADE. We have considered here only the needs of the most demanding pipelines and the following are clearly our best estimations as of today. In particular, while it is clear what will be needed to carry out CW searches over a given parameter space, the CBC needs will vary a lot depending on the number of the triggers found. New, unexpected results might clearly vary the scenario and hence the computing needs.

- **2014**
  Request: 1000 cores (power: 10 kHS06)

The CW group will be analyzing data from the Virgo run, exploiting smaller regions of the Sky and/or small frequency bandwidths for low values of τ_{min}. The CBC group needs to run tests and analysis on MDC, to optimize the algorithms in view of ADE. We have thus estimated a need for 1000 cores to be shared between the CW and the CBC groups. There is no need for additional storage.

- **2015**
  Request: 2000 cores (power: 20 kHS06)
  By this year the LIGO detectors will begin data taking. Three months of data will be available and thus new searches will begin. In parallel to this, the CW group will still be analyzing data from the Virgo run, exploiting smaller regions of the Sky and/or small frequency bandwidths for low values of τ_{min}. We have thus estimated a need for 2000 cores, to be shared between the CW and the CBC groups. We would need only an additional storage of the order of 20 TB in each CC, for the aLIGO data and some AdV data from commissioning. **Ask if there will be some important commissioning activity in 2015, with data to be stored**

- **2016**
  Request: 2000 cores (power: 20 kHS06)
  By this year aLIGO and AdV will run for 6 months. The CW group should have almost completed the main part of the analysis of past Virgo data and it will be too early to begin to analyze the new data (as this analysis needs to be done after having accumulated some, at least months, data). So some computing time will be needed to complete the analysis of past data and to begin first tests on the new data. The CBC group can begin real analysis. With 2000 cores they can analyze O(5) triggers in roughly 8 months (using the available nodes at 40%). We would need additional storage, as detailed in Table 8.7, scaled by the actual run time. Considering 6 months of commissioning and 6 months of science data, we would need roughly 1 PB on disk and 1 PB on tape.

- **2017**
  Request: 4000 cores (power: 40 kHS06)
  By this year aLIGO and AdV will run for 9 months. We have thus estimated that 4000 cores, again shared between the CW (to begin the analysis on the new data) and the CBC group, will be needed. We wouldn't need any additional disk space (above 1 PB) if already bought in the year 2016, and we would need 1 PB additional tape space to store one year of commissioning and science data.

8.8 Abbreviations

- **ADE** - Advanced Detector Era
- **AdV** - Advanced Virgo
- **aLigo** - Advanced Ligo
- **cWB** - coherent WaveBurst
- **CBC** - Compact Binary Coalesence
- **CC** - Computing Center
- **CM** - Computing Model
- **DA** - Data Analysis
- **DQ** - Data Quality
- **DS** - Data Segment
- **FFL** - Frame File List
- **GPU** - Graphical Processing Unit
- **GUI** - Graphical User Interface
- **GW** - Gravitational Wave
- **ITF** - Interferometer
- **IGWD** - Interferometric Gravitational Wave Detector
- **LDG** - LIGO Data Grid
- **LVC** - LIGO and Virgo collaboration
- **MDC** - Mock Data Challange
- **MoU** - Memorandum of Understanding
- **NM** - Noise Monitor
- **NMAPI** - Noise Monitor Application Programming Interface
- **RDS** - Reduced Data Set
- **WDF** - Wavelet Detection Filter
- **NoEMi** - Noise Event Miner
- **SILeNTe** - System Identification Linear et Nonlinear Techniques

Bibliography

Bibliography

[1] Virgo coll.
Virgo note VIR-0026B-12 (May 2012):
https://tds.ego-gw.it/itf/tds/file.php?callFile=VIR-0026B-12.pdf

[2] Virgo coll.
Virgo note VIR-PLA-DIR-7000-122 (Dec 2001):
https://tds.ego-gw.it/itf/tds/file.php?callFile=VIR-PLA-DIR-7000-122.pdf

[3] LSC and Virgo coll.
VIR-PLA-DIR-1000-223 (March, 2007):
https://tds.ego-gw.it/itf/tds/file.php?callFile=VIR-PLA-DIR-1000-223.pdf

[4] draft MoU (to be approved) LIGO-M060322-v3 and VIR-0324A-11

[5] LIGO Scientific collaboration (LSC)
LIGO-M1000066-v17 (Jan, 2012):
http://dcc.ligo.org/cgi-bin/DocDB/RetrieveFile?docid=9967&version=17

[6] LSC and Virgo coll.
VIR-067A-08 and LIGO-T970130-v1 (2009)
https://dcc.ligo.org/public/0000/T970130/001/T970130-v1.pdf

[7] LSC and Virgo coll.
VIR-0271A-12 (May 2012):
https://tds.ego-gw.it/itf/tds/file.php?callFile=VIR-0271A-12.pdf

[8] P. Astone, s. Frasca, C. Palomba

CQG 22 S1013 (2005)

[9] LV collaboration,
work in progress, the paper will probably be ready by the 2014 spring

[10] put here the reference to a document where the computing needs for PE and TIGER are estimated

[11] P. Jaranowski and A. Królak, Phys. Rev. D **61**, 062001 (2000).