

The Advanced Virgo Computing Model's Implementation Plan

Gergely Debreczeni

Foreword

A plan without breakdown of tasks, specification of timing or without manpower estimation will always remain a dream...

Content

- Philosophy of the Implementation Plan
- Project management
 - A project management tool
 - Teleconferencing issues
 - Web page issues
- Software maintenance
 - Revision control system
 - Build system
 - Software distribution, installation, repositories
- Data handling
 - Data management
 - Data access
 - Data transfer
 - Low latency data transfer
 - Bulk data transfers
 - File catalogs
- Job submission
 - Local job submission methods
 - Remote job submission framework
 - The Virgo Virtual Organisation
- Commissioning , DetChar, Data Analysis Workflows
 - Revision control system
 - Build system
 - Software distribution, installation, repositories
- Authentication and user credentials
- Computing awareness campaigns
- Manpower issues
- Difficulties
- Future steps

Philosophy of the Implementation Plan I

- The Computing Model is the abstract description of the computing workflow and related activities, actions, etc.
- The goal of the Implementation Plan is to map this abstract workflow to real-world computing infrastructures and to define, prioritize and schedule the steps, action items and their interdependencies which are necessary to achieve the goals specified in the Computing Model. This includes breakdown of tasks, costs and assessment of available manpower.
- In case of open questions, problems the implementation could / should contain several (even competing) solution and / or action item which have to be performed for the evaluation and testing of the alternative scenarios.
- During the actual implementation the tasks and their statuses must be followed up to ensure that no bottlenecks or dead-locks are formed in the execution of the tasks.
- The IP should be a living document only in the sense that it should contain pointers to places where the actual status of the tasks are updated.
- A single bigger document is better than a lot of small one, because it can give a more coherent view of various (sometime dependent) computing activities.
- The scope of the current Implementation Plan is a bit wider than that of the offline data analysis activities. and contains issues with highly varying priorities.

Some problems of Virgo Computing

- Computing solutions used sometime are years behind industry standards and some cases lagging behind LIGO solutions.
- Virgo computing is in some sense isolated, software solutions does not follow Linux standars, software packages are difficult to be used by external collaborators.
- Job submission solutions and frameworks are incompatible with LIGO solutions, LIGO colleagues cannot use our resources.
- Resources are too much heterogenous and fragmented.
- Lacking manpower for computing issues.
- Too much custom development instead of using out-of-the-box solutions available out there.
- In the last years Virgo users were using literally 100 (hundred) times !!! more LIGO computing resources than Virgo resources... while no Virgo resource is used by LIGO colleagues

The Implementation Plan is intended to give solutions to some of the above problems, but this requires a lot of work and changes in our way of computing... which requires - among other things - a clear definition of decision mechanism for computing issues which is currently missing.

Philosophy of the Implementation Plan III

Any specific choice solutions should be based on commonly agreed requirements

Philosophy of the Implementation Plan IV

Standardisation

Efficiency

Compatibility

Portability

Reliability

Project management I- A project management tool

- For any distributed computing project involving lot of participant in different administrative domains and countries, it is very useful to have a project management tool.
- After evaluating several possibilities, we have chosen Redmine for this purpose. Redmine is not only a bug tracker, reporter (such as SPR) but a more complex project management solution that in principle can be used by any project (not only computing).
- We've introduced to Redmine the various tasks of the Implementation Plan and they will be updated whenever their status changes
- In order not to duplicate the tools in the future Redmine could replace SPR but this is just an idea, needs careful evaluation and discussion before any kind of act.

Project management II - Teleconferencing

- In order to improve the communication, on VDASC calls we started to use TeamSpeak instead of SeeVogh. It is a lightweight client, easy to use, has much better sound quality.
- TeamSpeak has no screen sharing functionality, however it is less important that robustness and good audio quality.
- TeamSpeak runs on mobile devices

Project management III - Web pages

- Computing activities would need well organized, structured and up-to-date web pages. While investigating the situation we realized that this is valid for Virgo web pages in general.
- A serious web page reorganisation is needed for Virgo in general concerning content, style and content management system.

Software maintenance I - Revision control system

- A revision control system is needed for sure. We have been using CVS and recently changed to SVN. However SVN does not brought much benefit...and has some drawbacks (weak conflict resolution capabilities, missing concepts of frozen tags, etc..)
- The structure of the SVN repository itself needs significant clean-up. (For example documents and software should be better separated, some clarification on naming conventions)
- There is an ongoing discussion on moving to GIT, the question is still open.

Software maintenance II - Packaging, installation, configuration

- Currently CMT is used for configuration and distribution of the software. It works fine for Virgo internally, however it works only for Virgo....
 - It depends on SVN, does not support GIT
 - It is not compatible with Linux standards and with that of the LIGO way of software installation
 - User basis is decreasing (Atlas and LHCb is moving / has moved away) and long term support is not guaranteed (only 0.5 FTE)
- CMT has a lot of very nice feature which might not be trivial to substitute, but we should look for solutions which is used in the Linux world... and we should follow standards used by hundreds of thousands of other users and projects.... for example some tool based on CMake (just one possibility).
- This will require evaluation and testing of possible future tools and a lot of discussion and coordination between different people involved in software in Virgo.
- Standard (relocatable) Linux packages (.deb/.rpm/.tgz) and APT/YUM repositories would be more usable. Again, under discussion

Software maintenance III - Build system

- Currently Virgo doesn't have a build system, which is not good.
- Software packages doesn't have unit, regression and nightly tests.
 - Dangerous, question of reproducibility...
- A basic build system would be really useful, automatically running the tests and building the packages.
- Modern build systems are continuous integration servers, such as for example Jenkins
- Open question, not decided, requires time and manpower to set up

Data management

Data transfer - Bulk data transfer

- Bulk data transfer is the responsibility of EGO IT Department
- Together, we have investigated some alternative solution that might help in some aspect
- Different sites are using different protocols
 - Lyon - iRods - very convenient, robust, but used only in Lyon
 - CNAF - gridFTP - grid standard, used everywhere else, but requires more care to use
- To get the best support from the site, it is a good strategy to use the preferred protocol of the site
- The EGO developed transfer framework does exactly this
- This framework will be used for Advanced Virgo as well
- Some exercise has been defined to be performed in order to prepare for the data taking period

Data management

Data transfer - Low latency data transfer

- No information

Data management

Data access - Bologna, Lyon

- The local data access methods has been documented. No significant new developments are required or changes foreseen.

Data management

Data access - Bologna, Lyon

- The local data access methods has been documented. No significant new developments are required or changes foreseen.

Job submission framework

Local job submission

- The local job submission methods for the relevant CCs has been documented and are expected to be practically identical with the current solutions. Examples and descriptions provided.

Job submission framework

Remote job submission

- The development of a remote job submission framework is a new activity.
- It is necessary to make better use of our resources.
- There are multiple possibilities, each has advantages, disadvantages
- Dirac and Pegasus are the two most promising candidates
- There is a well defined set of steps necessary to perform before deciding which one to use
- In principle, the two can be used simultaneously, usage statistics will tell which is more useful for Virgo and LIGO

Job submission framework

An example for breakdown of task

6.3 Verification and testing process for the job submission framework of candidates

Here we define the necessary steps to be performed before making the final selection of the distributed job submission framework. These steps are

1. (rt#9, rt#27) Contact the developers and get a statement about the future development roadmap
2. (rt#10, rt#28) Check whether the software is available for the selected OSes and platforms of AdV
3. (rt#11, rt#29) Download, install and configure the software
4. (rt#12, rt#30) Check which computing resources can be reached / enabled with the given solution
5. (rt#13, rt#31) Check job management capabilities, such as submit, remove, stop, hold, logging, etc..
6. (rt#14, rt#32) Check up-to-dateness of documentation
7. (rt#15, rt#33) Check the active user bases, forums, support possibilities
8. (rt#16, rt#34) Try to set up and submit a "Hello world" job for various resources.
9. (rt#17, rt#35) Examine the user experience of the software
10. (rt#18, rt#36) Try to submit a more complex workflow
11. (rt#19, rt#37) Test data access from within the submitted job
12. (rt#20, rt#38) Test file catalog, job sub submission system interaction if there is any

Job submission framework

File catalogs

- Different job submission frameworks are requiring different file catalogs
- It would not be wise to multiply the number of different file catalogs
- The question is still open, we have to see which file job submission framework will be used

Job submission framework

Grid information system

- With the extended usage of the Grid we have to have a mechanism in place which checks the sanity of the information published by different Grid sites
- This information system can be used to find, allocate and distinguish between resources
- Have to be in sync with Pegasus site catalog

Job submission framework

The Virgo Virtual Organisation

- For the remote job submission framework the user has to use the grid certificates
- These certificates have to be registered in the Virgo VO server
- The usage and maintenance pattern of the Virgo VO server will probably continue in the very same way as it is currently done
- No need for any new action
- Will have to check how to register LIGO colleagues when / if they will try to use the distributed job submission framework
 - Problem: The CA used by LIGO people are not recognized by the EUGridPMA

Detector characterisation

- Data quality: Omicron pipeline, On-line vetoes, Detector Monitoring System (DMS), MonitoringWeb, Spectrograms, DQ developments, DQ flag reprocessing, DQSEGDB
- Noise studies: NMAPI, NoEMI, SILENTE, Regression, WDF, Coherence, Nonstationary monitoring

The above pipelines are documented, computing related actions are mentioned.

Scientific data analysis workflows

- CBC: MBTA, Testing GR, GWTools, CBWaves pipelines/tools
- CW: Rome PSS, Targeted search, Polgaw pipeline, GWTools

The above pipelines has well understood development roadmap documented in the IP.

- Burst: No information
- Stochastic: No information

Authentication, autohorization

- We need more uniform way of authenticating users againts various clusters, services
- Too much different credentials to handle, to know
- Authenticaltion of Virgo users for LIGO services and vica versa
- Future: SAML based solutions, identity providers, federations
- Web service / host certificates used should be commercial ones instead of self-signed

Computing awareness campaigns

- In order to help data analysis people to create more efficient analysis software a series of tutorial will be held (is hold) on various topics, such as
 - remote job submission, grid stuff
 - GPU computing
 - paralellisation techniques

Manpower issues

- Always an issue
- Too few people has time for computing
- Very very few student
- We need dedicated manpower for computing issues which are not directly belonging to any of the data analysis pipeline / group

Difficulties

- First time ever done for computing activities
- Partial success
- Misunderstanding of concept of „tasks”
- The documentation forms for various activities are different, should be better integrated