Data Intensive processing with iRODS and the middleware CiGri for the Whisper project

Briand X.* Bzeznik B.†

Abstract

blabla

blabla

Keywords

Data-Intensive, grid computing, distributed storage, Seismic Noise, Whisper, Cigri, Irods.

1 Plan

1. Introduction, abstract (3§, 1 page)
2. Background (7-10§, 1-2 pages)
   (a) The Whisper project
   (b) Size of observational data
   (c) Size of computational data
   (d) General Big data/ science of universe
   (e) Coupling scientific objectives / IT constraints
   (f) Data grid paradigm
   (g) Data grid in Grenoble
   (h) Collaboration Whisper/IT infrastructure
3. Software for data-intensive processing (6-12§, 1-2 pages)
   (a) General IT codes whisper
   (b) Computer language and librairies
   (c) Structure of codes
   (d) Package for raw data
   (e) Package for correlations
   (f) First optimisation of correlation
   (g) Second optimisation of correlation
   (h) Codes for analysis of correlations
   (i) Where we can run the codes
4. IT infrastrcuture for grid computing (6-12§, 1-2 pages)
   (a) Needs of coupling storage computation
   (b) Presentation Ciment
   (c) Coupling Irods/Cigri
   (d) Resulting Data-Grid
   (e) Presentation Irods
   (f) General Infra Irods (diagram)
   (g) Effective Nodes Irods (diagram)
   (h) General Cigri
   (i) Mechanism with OAR
   (j) Mechanism resubmission/besteffort
   (k) Description of a campaign
   (l) Low Interaction Cigri/Irods
   (m) Cigri V2 and V3 functionalities
5. Results (28-31§, 6-7 pages)
6. Discussion (7-10§, 1-2 pages)

*Isterre, Cnrs, email xav.briand@gmail.com
†Ciment, Université Joseph Fourier

2 Background 7-10§, 1-2 pages

The *Whisper* * project is a an european project on seismology whose goal is to study properties of the earth with the seismic ambient noise such that evolution of seismic waves speed. This noise is almost all the signal continuously recorded by the seismic stations worldwide (Europe, China, USA, Japan), except earthquakes. It offers new observables for the seismologists, new types of virtuals seismograms that are not only located at the place of earthquakes. For instance, one can obtain wave paths that probes the deepest part of the Earth.

Accordingly, this is one of the first project in the seismological community that studies systematically the continuous recordings, which represents a large amount of seismological data, of the order of several tens of terabytes. For instance, one year of the Japanese Network is about 20 TB or 3 months of the mobile network USArray represents 500 GB (it depends on the sampling of the recorders).

In addition, the calculation operations downstream may produce even more data than the observation data. To give an order of magnitude, more than 200 TB has been managed by the Whisper project at the same time. A classical processing produces 8 TB in 5 days. An other one 'read' 3 or 4 TB and 'produced' 1 TB in 6 hours. Many tests of the signal processing are done and computational data are deleted as and when required.

Nowadays, the earth sciences or more generally, sciences of the universe are widely engaged in data-intensive processing. This leads to design scientific workflow, towards data-intensive discovery and e-Science.

Reflected by the Whisper project, we have to organize the science objectives with the computer constraints. We have to take into account the duration of postdocs and PhD theses, as well as the availability of computer infrastructures and their ease of access. This leads to many questions about software development, including the genericity of computer code and the technical support. But it also influences in terms of choice of appropriate infrastrucures.

Even if this project has his own ressources, such a problem of data-intensive requires specific tools able to organize distributed data management and acces to computational ressources: a data grid environment.

The University of Grenoble offers, thanks to the High Performance Computing (HPC) centre *Ciment*, this kind of environement with the distributed file system *Irods* and the middelware *CiGri*.

It is thanks to the close collaboration between IT ressources of Whisper and the infrastrucrures of the University that this project has been implemented as we show below.

*FP7 ERC Advanced grant 227507, see whisper.obs.ujf-grenoble.fr

3 Software for data-intensive processing (6-12§, 1-2 pages)

A part of the Whisper project is specificaly dedicated to the IT codes. This includes to design a specification, an implementation and some optimisations of a sequence of a data-management and of a computations [†]. This project uses some own IT resources (servers, dedicated bay) but also uses common IT infrastructure of the university. We developed also adaptations for the IT infrastrutures and we provide technical support for researchers.

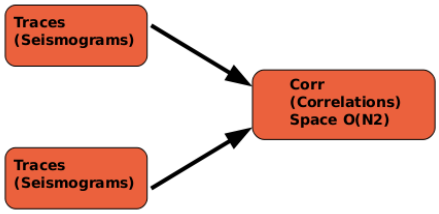
Most of the IT codes are writing with the *Python* language and uses intensively the scientific libraries *Scipy* (fortran and C embeded) and *Obspy* [‡] (essentially the 'read' function). The latest one is dedicated to the seismological community.

The IT codes consists of several tools described schematically at the figure 2 and grouped into three parts. The first one concerns the signal processing, the second part permits the computation of the correlations and the last part consists of codes for the analysis of the correlations.

A first package provides a flexible way to process raw data, to specify a pipeline of pre-processing of signal. The user starts by specifying a directory, a set of seismic stations and a set of dates. Then the codes scan the directory and extracts all pieces of seismograms (also called traces) and rearranges them in a specific architecture of files in order to calculate the correlations to the next step. We use here intensively the function 'read' of the library Obspy which allows to open most of format files seismogram. The user also define his own sequence of processings. He can use the functions predefined but also the Python libraries he needs. Moreover he can add eventually his own processing.

The second package concerns correlations. Roughly speaking, a correlation is an operation with two seismograms that provides the coherence between the both signal (for a given time windows). Moreover, it represents, in some favorables cases, a new virtual seismogram (it converges to the Green's function). Thus, the code computes all the correlations and provides an architecture of files that corresponds to all the couples of seismograms (for each date).

Figure 1: Step of the correlations



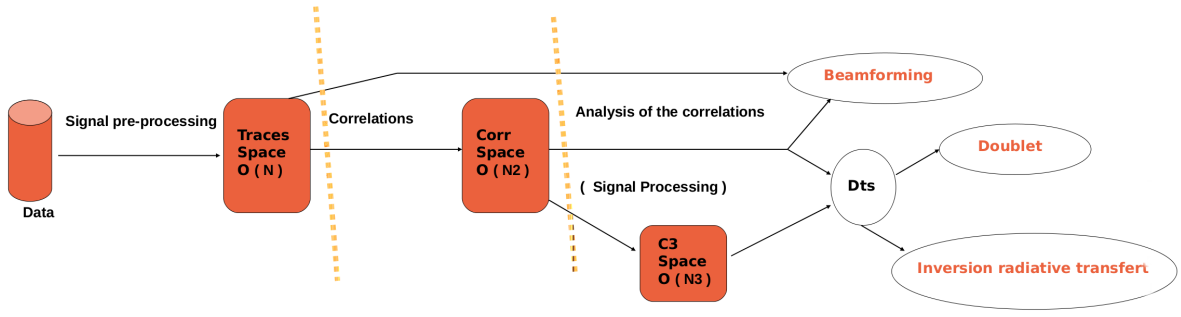
These quadratic space complexity can be critical and lot of effort was made in order to optimize the computation in two direction. First we improve the computation of the fast fourier transform by pre-calculating some "good" combinations of small primes numbers. With this method, we improve of forty percent the time computation in the favorable cases.

Nevertheless, the main optimization was made by testing the behaviour of the carbage collector of Python in order to follow the cache heuristics. More precisely, we do not use the gc module or the 'del' statement but we try to schedule and localize the line of code in order to find the good unfolding that uses the architecture optimally.

The last part of computer codes concerns the analysis of correlations (the virtual new seismograms) with methods such as beamforming, doublet or inversion. We also compute correlations of correlations C3 (also new seismograms). For example, we study the variations in velocity of seismic waves as we illustrate below.

These codes permit to process a dataset on a computer laptop. Nevertheless, to take advantage of IT infrastructure at the University of Grenoble, adjustments have been made for the grid computing as we shall see later.

Figure 2: Main sequences of processings of the Whisper Codes



[†] see code-whisper.isterre.fr/html/ (part of the design)
[‡] see www.obspy.org

4 IT infrastrcuture for grid computing (6-12§, 1-2 pages)

The data-intensive processing needs obviously an IT infrastructure in order to couple storage and computation. In our cases, most of the processing are embarrassingly parallel. The amount of data and the location of compute nodes available suggests using a distributed storage system with a grid manager.

The IT infrastructures used here are provided by the *Ciment* [§] platform. Ciment is the Grenoble University High Performance Computing (HPC) center. It offers a partial pooling of computing (6600 cores, plus GPU, 10 clusters) and many documentations for users. Moreover, the computational resources are integrated in a local grid, a grid of supercomputers. Associated with a distributed storage, it provides a local data grid environement.

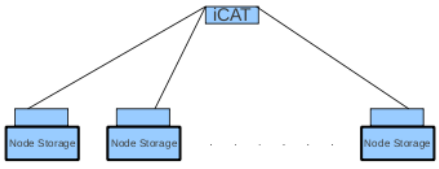
The distributed storage accessible by all the clusters is established and it is managed by iRODS [¶]. Nowadays it represents approximately 700 TB. Moreover, a grid management, the *CiGri* [‖] *middleware*, allows both access to iRODS storage and access to computing nodes of the clusters of the university (with the resource manager OAR [**]).

With Irods and CiGri, we can then access to the computational power of clusters of the University of Grenoble in mode best effort (with grid or parametric jobs). It may be noted that iRODS also acts to the user as a centralized controller with a total observation and thus allows the user to control its calculation.

The Integrated Rule-Oriented Data System (iRODS) is a data management software. It permits to manage data independently the ressource storage with the i-commands or the iRODS API. It allows also the user to define metadata associated to the data (Collections) and make query on them (see [1] for an illustration of use). Moreover, it is a glass box, it is possible to define rules that call services (called microservices). One can define then automatic replication between differents ressources or more generally a processing for a selected data.

The iRODS storage infrastructure of Ciment consists of a zone with the iCat server and several nodes, of the order of several tens, see figure 3.

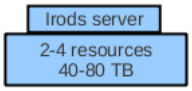
Figure 3: iCAT and nodes storage



[§] see ciment.ujf-grenoble.fr
[¶] see irods.org
[‖] see ciment.ujf-grenoble.fr/cigri/dokuwiki
[**] see oar.imag.fr

Capacity has now reached 700 TB and is constantly evolving and increases with investment in new projects. Each node comprises an Irods server coupled with further ressources as illustrated at the figure 4

Figure 4: A node storage



There is also a web interface where the user can check the status of the ressources (figure 5). The resources are grouped into three sites as near as possible of the supercomputers. Note that one of the sites is 4km from the other two.

Figure 5

The access to 6600 cores of the clusters of the Ciment platform is achieved through the middleware CiGri. Cigri launch jobs on idle processors of every computing clusters and then can optimize the resources usage.

Each cluster of the University of Grenoble uses the resource manager OAR. Cigri acts, among other things, as a metascheduler of OAR. It retrieves the clusters states trough OAR and submits the jobs on free resources.

While it may work in normal mode, CiGri is used in BestEffort mode, ie that its jobs are lower priority: they can be killed by any other job. However, CiGri provides automatic resubmission. With this mecanism, the user can submit a big amount of small jobs, called a campaign, and he didn't care anymore.

Roughly speaking, in order to run a campaign, the user describes through a file (in json format) the parameters of the campaign such as the accepted cluters, the maximum duration, the location of the codes, and a prologue on each cluster in order to retrieve data and codes on iRODS. Moreover, it define also a file where each line correponds to a value of the parameter for the user's code. Thus, the number of line of these parameter file corresponds to the number of jobs of the campaign.

Note that iRODS is accessible by all the nodes of all the clusters. By this way, for a given campaign, CiGri gets not only the data to be process but also the codes to be run on

the node. CiGri is therefore very independent of the data management of iRODS.

CiGri is now at the version 3, which represents a major evolution in terms of modeling and technology (Rest API, Ruby)

5 Results (28-31§, 6-7 pages)

References

[1] Gen-Tao Chiang, Peter Clapham, Guoying Qi, Kevin Sale and Guy Coates, Implementing a genomic data management system using iRODS in the Wellcome Trust Sanger Institute. *BMC Bioinformatics,12(1):361+, September 2011.*