Low Latency transient search of Gravitational Waves for the Advanced Detectors

M. Drago[1]

[1] Max Planck Institute for Gravitational Physics (Albert Einstein Institute), Callinstrasse 38, D-30167 Hannover, Germany and Leibniz Universiat Hannover, Welfengarten 1-A, D-30167 Hannover, Germany

(Dated: July 13, 2015)

The reliability of the first detection is one of the most interesting challenges for the gravitational wave community. To increase the detection confidence, the LIGO and Virgo collaboration have already started coincident observations between gravitational waves detectors and other astronomical instruments, like electromagnetic or neutrino detectors. This can be done in two directions: searching for gravitational waves triggered by the electromagnetic informations, or pointing the electromagnetic telescopes to the sky position given in real time by the gravitational wave analysis. The success of the latter case depends strongly on the analysis speed of gravitational wave pipelines to analyze data and extract any gravitational wave candidate with as much information as possible. In this paper we discuss the case of the coherent Waveburst pipeline, the main pipeline used in the past scientific LIGO-Virgo analyses for the search of gravitational wave transients, reporting the capability of making an all-sky and all-time analysis and the analysis speed performance.

PACS numbers: 04.80.Nn, 07.05.Kf, 95.55.Ym, 04.30.Db

I. INTRODUCTION

The gravitational wave (GW) community has achieved significant progresses towards the search for GWs, thanks to the innovative operation of the Laser Interferometer Gravitational Wave Observatory (LIGO) [1] and Virgo detector [2]. The era of gravitational wave astronomy is going to start in the next years, with the starting of Advanced LIGO and Advanced Virgo, as we expect to detect numerous GW signals before 2020 [3, 4].

Transient gravitational waves can be produced by a wide range of astropysical processes (compact binary sistem merging [5, 6], core-collapse supernovae [7], neutron star collapse to black holes [8]), for most of them the we do not have a precise model of the GW signature. This forces to implement a search that must be sensitive to the widest possible variety of waveforms.

This un-modeled search has the great disadvantage that it is difficult to distinguish a possible true signal from noise glitches that mimic a true GW (false alarms), even if coherent searches already significantly reduce the number of false alarms with respect to single detector or time-coincidence searches.

To assign a more reliable confident to any GW detection, a coincidence observation with electromagnetic (EM) or neutrino counterpart would be an interesting approach. Moreover, joint multimessenger observation would bring a more complete information on the source, like idenfitication of host galaxies or the unveal of its inner dynamics.

A variety of GW emission processes are likely to be associated to also EM emission, like: Gamma-Ray bursts (GRBs) [9–11], or merger of two compact objects leading to a supernova-like transient [12]

The LIGO and Virgo collaborations has already performed GW searches associated with other astrophysical manifestations (Gamma-Ray Bursts [13, 14] or neutrino [15, 16]). The idea is to restrict the GW search around the time and sky position given by the partners (*ex-triggered*). This has a natural advantage: information from external triggers of sky position and arrival time allows to make a specific targeted search. This naturally reduces the rate of false alarms, simply because less data are analyzed, both in time and sky area. Moreover, restricting the parameters space allows to assess more confidence on the eventual detected GW trigger. [17]

In the last scientific run (2010), the LIGO and Virgo communities developed the *follow-up* procedure: GWs become the triggers for other astrophysical experiments. To allow this possibility, have been implemented algorithms able to make a fast search in real time and reporting information to EM partner so to point telescopes to the directions in the sky given by the GW alerts [18, 19].

The success of this approach depends on two capabilities of the considered algorithms: make an accurate sky locations of the GW and get a enough low-latency between the trigger arrival time and the alert to the EM community.

Studies on sky localization accuracy have been already performed in a huge variety in literature: from analytical studies [21–26] to applications of coherent netwok analyses [27–29].

In this paper we consider the coherent WaveBurst (cWB) algorithm [30]. Sky localization accuracy for this algorithm has been already reported in [31] and [32]. Here we describe how the cWB is structured to allow a fast alert to EM partners.

The paper is organized as follows: in Sec II we describe the algorithm characteristics, and the adopted solutions to allow a fast search, in Sec III we shows the performances of the algorithm on the last LIGO engineering run.

II. SEARCH ALGORITHM

A. Offline search

The Coherent WaveBurst (cWB) algorithm[30] has been already used for the search of transient signals in the past scientific run of LIGO and Virgo collaborations [33, 34]. A new version of this algorithm (cWB 2G) [35] has been developed in preparation of the Advanced Detector Era.

cWB is c++/ROOT[36] based excess power algorithm that combine the data coming from a network of detectors calculating the maximum coherent likelihood along a discrete grid in the sky. First of all, it applies a time-frequency (TF) transformation [37] at different TF resolutions, so to adapt the TF transform to the characteristic of the signal. For instance, a signal which is well localized in frequency, is better described by high frequency resolution TF transformation. Instead, a signal which covers a large frequency band, is better described by low frequency TF transform. Then cWB selects from the TF transforms of the detector data the pixels with energy above an adaptive threshold (depending on the noise level). These selected pixels are collected in a unique cluster if they satisfy some "neighbours" rule. Then cWB applies Principal Component analysis on all the TF resolution to find what is the optimal set of pixels that is more adapted to the signal TF characteristics. A typical coalescence signal, for instance, should be described by TF pixels with higher frequency resolution during the coalescence stage, while the post-merger stage should be described by higher time resolution.

For each of these clusters, the pipeline choose the estimated sky location from a probability sky map depending on the likelihood, and for such sky location all the event parameters are calculated according to the chosen sky position. Then an event is selected if it satisfy some internal threshold, i.e. the network correlation coefficient and the correlated amplitude [33, 34]. The significance of the trigger is compared to the background estimation performed using time-slides between detectors. Time-slides are performed for each segment in a "circular" way, this means that if we have the segment $[t_0, t_0 + T]$ and we are making a shift of length t of the first detector with respect to the second, the time period $[t_0 + T - t, t_0 + T]$ of the first detector is considered "coincident" with the time period $[t_0, t_0 + t]$ of the second detector (Fig. 1).

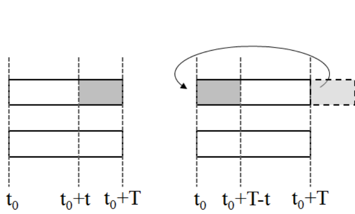

FIG. 1: *Simple visualization of circular lags.*

To perform time-shifts greater than the segment length, it is possible to consider "coincident" a segment $[t_0, t_0 + T]$ of the first detector with a segment $[t_1, t_1 + T]$ of the second. We call this feature super-lags (Fig. 2).

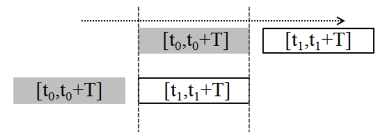

FIG. 2: *Simple visualization of super-lags.*

This is the standard algorithm that is used for the high-latency, which is usually performed at the end of a scientific run data taking ("offline"). The main information required from the analysis are: the frame files (FR) containing detector data and data quality (DQ) files of analyzable times. These information are usually given by the scientists during or at the end of the run. In the low-latency analysis the algorithm should get these informations directly in real time.

B. Online infrastructure

The online cWB infrastructure is a set of python code that extracts the available information as soon the FR are reported on the cluster machines, selects the analyzable periods, launchs the cWB pipeline on these periods to extract the triggers and sends the alerts to a dedicated database.

Usually FR contains four seconds of data and the information about their proper start and duration can be extracted directly from the file name. The FR contains, in addition to the GW data channel, a DQ channel recording the time periods that can be analyzed. The pipeline extract the list of interesting time periods and wait for a minimal amount of continous available time. When this amount is reached, it prepares the standard configuration of the offline analysis for this segment and launch the analysis. The minimal amount should be decided to be great enough to allow a safe application of linear predictor filter (an algorithm that clean the persistent noise with high energy that characterize each detector, produced by enviromental and electronical issues [38]) and whitening. More this minimal value is smaller, faster is the pipeline to analyze it, and giving information on the extracted trigger. Unfortunately, this increase the probability to loose signals that can be near the segment border, or between two continous segments. To avoid this issue, the pipeline can afford multiple analysis instances that are shifted among each other less than the segment length. Then the pipeline compare the results from each instance, counting only once the triggers that comes up from the multiple analyses, so not to give more times the same trigger to the EM partners.

At the same time, the algorithm perform time-slides on the data to assess a confidence on the detected triggers. This step is crucial because ideally we would like to have a huge number of time slides to assess the trigger confidence with a high significance. However, we are limited by two main issues: duty cycle and computational load. For two detector cases we need a minimal continouos time period of coincidence data to perform a certain number of time-slides. For instance, with a time step δ to perform N time slides, we need at least a continuous coincidence time of lenght $N\delta$, that it is less probable as more time slides we planned. Fortunally, the introduction of super-lags allow to perform analyses considering segment with a length T that is less than the requested time, so to decrease the computational load for each analysis process. Indeed, we are using in this case $M = N\delta/T$ processes, each performing $L = N/M$ lags. However, more time slides we are performing, more computing time we need to run the analysis. A possible solution is to split the number of time slides to small subset, so to run in parallel K instances of less time-slides each. This means that we need $M \cdot K$ different machines if we want to perform as fast as possible the total N set of time-slides. The length T, the number M and K are free parameters that can be decided to obtain a good compromise between the desired N lags, the running time of each job and the number of total jobs (Fig. 3).

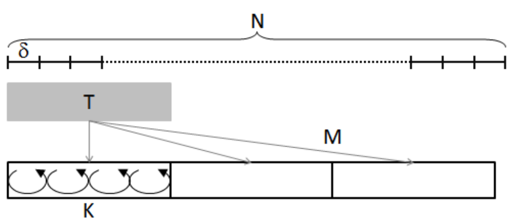

FIG. 3: *Simple visualization of lags splitting in different analyses to reduce the computational load of the pipeline.*

III. PERFORMANCE

The most important value to quantify the performance of the algorithm is how much time it needs to send an alert after the instrinsic trigger time (*alert time*). However, we can identify different processes that leads to this final value.

First of all the time needed to read the frame files and launching the algorithm on a given segment (job). This is defined as the difference between the GPS time when the job effectively starts running and the GPS time of the segment end (*delay launch*).

Secondly we are interested in the effective job *running time*, i.e. the time between the jobs effectively starts and stop running. Moreover, this information is completed by the *finished time* which takes in account also for the other instances to finish. This quantity reports the time from the run start and the end of comparison procedure for the eventual triggers coming from the two instances.

The total sum of all these factors is resumed in the *completion time*, i.e. the time between the end GPS of the segment and when the comparison procedure is done. The alert time of each trigger is related to the completion time of the related job where we have to add the time distance between the trigger time and the job GPS end.

The algorithm has been applied during the seventh LIGO engineering run (June 2015): about 2 days of coincidence data considering the LIGO detectors only (Livingston and Hanford). This is the expected configuration when the two Advanced Detectors will become online for the first scientific run (September-December 2015). All of the statistics considered in this work refers to this particular run.

The analysis consisted of two instances of 60 seconds length shifted of 30 seconds.

The delay launch time can be from one to two minutes, as reported in Fig. 4

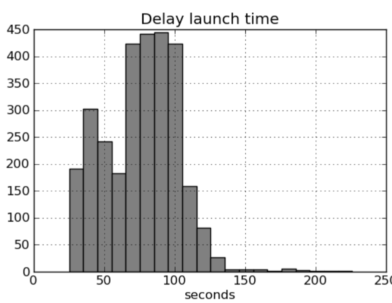

FIG. 4: *Launch time distribution: Time between the running start and the GPS segment end.*

The running time of each single 60 segment depends strictly from the noise level of the data, but on average the pipeline finishes one job in about 34-37 seconds for both instances (Fig. 5).

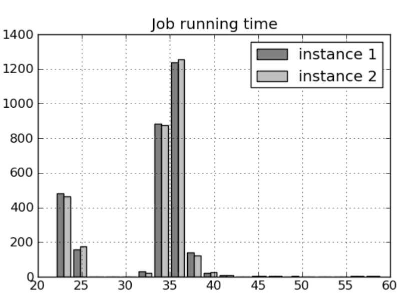

FIG. 5: *Running time distribution of the two instances of 60 second jobs.*

If we add the time for the trigger comparison, the finished time shows a mean value around 40-60 seconds (Fig. 6.

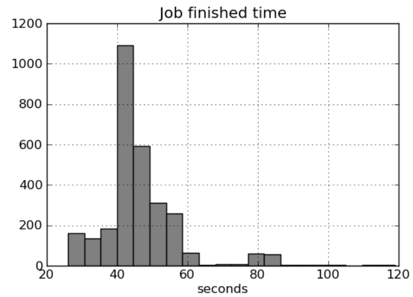

FIG. 6: *Finished time distribution: time between the starting of the main instance job and the comparison procedure.*

Finally the total completion time is for most of the segments less than 3 minutes (Fig. 7).

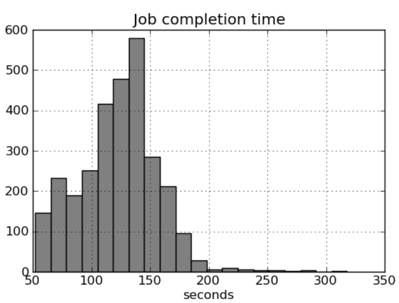

FIG. 7: *Completed time: time between the GPS segment end and the comparison procedure.*

From these we expect that the alert time is at least five minutes after the trigger intrinsec time.

For assessing significance on the triggers we made 1000 lags ($N = 1000$) using two super-lags of 600 second segments ($T = 600$, $M = 2$) with 5 instances of 100 lags each ($K = 5$).

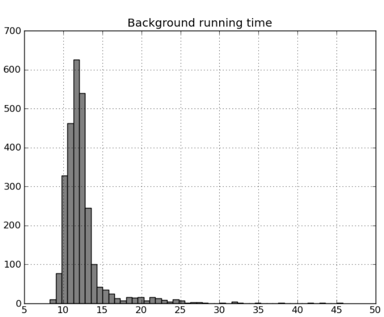

FIG. 8: *Running time of the background jobs.*

Most of background jobs of 100 lags run in 12-15 minutes (Fig. 8). These jobs were distributed in parallel among ten different machines, so the multiple jobs of 100 lags each run approximately at the same time.

IV. CONCLUSIONS

We resume in the Table I the speed performances of the various step of the analysis, as explained in Sec III.

Stage		Mean	Std	Min	Max
Delay Launch Time	[s]	76	26	25	226
Running Time 1	[s]	32	5	22	59
Running Time 2	[s]	32	5	22	58
Job Finished Time	[s]	46	11	26	119
Job Completion Time	[s]	122	33	52	318
Time slides	[m]	12.4	2.9	8.2	45.8

TABLE I: *Statistics about speed performances of the different stages of the analysis, reporting in the various columns the mean (Mean) and standard deviation (Std), and the minimum (Min) and maximum (Max) values for each step.*

This results are similar to the obtained during the last scientific run [39, 40], where the alter latency was around five minutes. This is a short time compared to the human validation step, a collection of consistency checks which decides if the alerted trigger should be send to the EM partners. This validation occurs around thirty minutes during the last run [39, 40], this means that the pipeline alert latency is risible with respect to the total alert. Anyway, making this alert faster will reduce the total process. It is possible to reduce the latency discarding the waiting for both instances to finish: a trigger found from instance 1 is send as soon it is detected, if the same trigger is detected with a bigger significance from the instance 2, it is simply substituted on the alert database. This will reduce the total time of around the difference between Finished time and the Running time. We are also investigating if we can reduce the delay launch time optimizing the extraction of information from the frame files.

The cWB online algorithm is adapted to run for low-latency analysis in the search of gravitational waves for transient signals. We demonstrated that the all pipeline infrastructure is able to be enough fast to alert the electromagnetic partners in some minutes after the incoming of the triggers. We can say that the pipeline is ready for the upcoming era of gravitational wave astronomy, when the Advanced LIGO will be online, and the Advanced Virgo detector will join the search in the next year.

References

[1] B.P. Abbott et al., Rep. Prog. Phys 72, 076901 (2009)
[2] F. Acernese et al., Class. Quantum Grav. 23, S635 (2006)
[3] J. Abadie et al, Class. Quantum Grav. 27, 173001, (2010)
[4] J. Aasi et al, arXiv:1304.0670
[5] F. Pretorius, arXiv:0710.1338
[6] Z.B. Etienne et al., Phys. Rev. D 77, 084002 (2008)
[7] C.D. Ott, Class. Quantum Grav. 26, 063001 (2009)
[8] L. Baiotti et al., Class. Quantum Grav. 24, S187 (2007)
[9] C.S. Kochanek and T. Piran, ApJL, 417, L17 (2003)
[10] S. Kobayashi and P. Meszaros, ApJ, 589, 861 (2003)
[11] J. Abadie et al. ApJ, 760, 12 (0212)
[12] L. Li and B. Paczynski, ApJL, 507, L59 (1998)
[13] LIGO-Virgo Collaboration, Astrophys. J. 715 1438 (2010)
[14] LIGO-Virgo Collaboration, Astrophys. J. 760 12 (2012)
[15] ANTARES Collaboration and LIGO-Virgo Collaboration, JCAP 1306 008 (2013)
[16] IceCube collaboration and LIGO-Virgo Collaboration, Phys. Rev. D 90 102002 (2014)
[17] M. Was, P.J. Sutton, G. Jones, I. Leonor Phys. Rev. D 86, 022003 (2012)
[18] J. Kanner et al, Class. Quantum Grav. 25, 184034 (2008).
[19] LIGO-Virgo Collaboration, ApJS 211 7 (2014)
[20] LIGO-Virgo Collaboration and Swift collaboration, ApJS 203 28 (2012)
[21] P. Jaranowski and A. Krolak, Phys. Rev. D 49, 1723 (1994)
[22] J. Markowitz, M. Zanolin, L. Cadonati, and E. Katsavounidis, Phys. Rev. D 78, 122003 (2008)
[23] S. Fairhurst, New J. Phys. 11, 123006 (2009)
[24] S. Fairhurst, arXiv: 1205:6611 May 30 (2012)
[25] L. Wen and Y. Chen, Phys. Rev. D 81, 082001 (2010)
[26] B.F. Schutz, Class.Quant.Grav. 28, 125023 (2011)
[27] Y. Gürsel and M. Tinto, Phys. Rev. D 40 (1998)
[28] É.É. Flanagan and S.A. Hughes, Phys. Rev. D 57, 4566 (1998).
[29] S. Klimenko, S. Mohanty, M. Rakhmanov and G. Mitselmakher, Phys. Rev. D 72, 122002 (2005)
[30] S. Klimenko et al, Class. Quantum Grav. 25, 114029 (2008)
[31] S. Klimenko et al.m, Phys. Rev. D 83, 103001 (2011)
[32] R. Essick et al., arXiv:1409.2435
[33] J. Abadie et al, Phys. Rev. D 81, 102001 (2010)
[34] J. Abadie et al, Phys. Rev. D 85, 122007 (2012)
[35] S. Klimenko and G. Vedovato, in preparation
[36] https://root.cern.ch/drupal/
[37] V. Necula et al., J. Phys.: Conf. Ser. 363, 01203 (2012)
[38] W. Tiwari et al., arXiv:1503.07476
[39] J. Aasi et al., The Astrophysical Journal Supplement Series 211, 7 (2014)
[40] J. Abadie et al., Astronomy & Astrophysics 539, A124 (2012)