INSTITUTE OF PHYSICS PUBLISHING CLASSICAL AND QUANTUM GRAVITY

Class. Quantum Grav. **20** (2003) S655–S664 PII: S0264-9381(03)62146-5

The search for continuous sources in the Virgo experiment. Full-sky incoherent step: 'local' and 'grid' tests

L Brocco, S Frasca, C Palomba and F Ricci

Università di Roma, 'La Sapienza' and INFN Roma1, Ple A Moro, 5, 00185 Roma, Italy

E-mail: cristiano.palomba@roma1.infn.it

Received 14 April 2003
Published 7 August 2003
Online at stacks.iop.org/CQG/20/S655

Abstract

The full-sky incoherent step is the most computationally heavy in the hierarchical procedure developed in the search for continuous gravitational signals in the data of the Virgo detector. This step is based on the Hough transform. We have implemented it by means of an approach that uses look-up tables. Here, after a short introduction to the whole data analysis method, we describe the implementation of the Hough transform and discuss the performances of the code. Then, we discuss the computing framework in which data analysis will be performed. In particular, we briefly describe the architecture of the European DataGrid software, which we have used to deploy a small computational 'grid'. In this 'grid' environment we have tested our code and the results are shown.

PACS numbers: 04.80.N, 07.50.Bx, 07.05.Kf

1. Introduction

The detection of periodic gravitational signals emitted by astrophysical sources is one of the main goals of today's and forthcoming detectors. According to current models about 10^9 neutron stars are expected to exist in our Galaxy. Most of these are probably rotating at a rate too low to be inside the frequency band accessible by detectors. A non-negligible fraction, however, is rotating at higher frequencies and then could emit detectable signals. The simplest model of a periodic source is given by a bi-axial rotating neutron star, asymmetric with respect to the rotation axis. In this case the frequency of the signal is twice the rotation frequency of the star. The detected frequency is not constant due to the intrinsic source spin-down and the Doppler effect due to the Earth's motion. The spin-down is a consequence of the loss of energy of the star through the emission of electromagnetic and gravitational waves; this last causes a slow-down at a rate proportional to the fifth power of the rotation frequency. The Doppler effect is the sum of two components: the Earth's orbital motion around the Sun, with

Table 1. Main quantities for the optimal analysis. The numbers refer to a total observation time $T_{\rm obs} = 4$ months, a frequency band up to $f_0 = 500$ Hz, a minimum neutron star decay time $\tau \equiv \frac{f_0}{\dot{f}_0} = 10^4$ years. A full-sky search is also assumed. $L_{\rm FFT}$ is the length of FFTs (number of points), $N_{\rm tot}$ is the total number of points in the parameter space and depends on the number of points in the sky, on the number of spin-down parameters to be taken into account and on the number of different frequencies to be investigated. CP is the computing power needed, $h_{\rm min}$ is the 'nominal' sensitivity, that is the amplitude of the signal, at the detector, that could be detected with a SNR equal to 1. In the sensitivity evaluations a detector noise power spectral density $S_h = 10^{-46}\,{\rm Hz}^{-1}$ is assumed.

$L_{\rm FFT}$	$N_{\rm tot}$	CP (Tflops)	$h_{\rm min}$
1.0×10^{10}	6.5×10^{28}	1.5×10^{16}	6×10^{-27}

a period of one year, and the Earth's rotation, with a period of one sidereal day. On timescales of the order of one day, or shorter, the latter is dominant. Other effects can further complicate the signal:

- *glitches*, which are sudden changes in the rotation frequency and its derivative, probably due to cracks in the neutron star crust or to the interaction between the crust and the inner superfluid;
- the presence of a companion star which adds a third component to the Doppler effect due to the orbital motion of the source in the binary system;
- the accretion of matter from a surrounding nebula, or another star, which can make the frequency evolution of the source not easily predictable;
- the case of a triaxial neutron star, which produces a splitting of the spectral lines and complicates the analysis.

In the following discussion we will not take into account these complicating factors and assume that the only relevant effects are the source spin-down and Earth's Doppler effect.

2. The detection of periodic signals

The basic method in the search for periodic signals of unknown frequency in the data of gravitational detectors consists in estimating the power spectrum using FFTs. Due to the weakness of the sources the signal must be integrated for a very long period of time, at least of the order of months, and this implies the calculation of very long FFTs. Moreover, due to the frequency variations described above, the signal is spread among many frequency bins. A way to achieve optimal detection in the case of varying frequency is to correct the data for the supposed position, frequency and spin-down of the source. In the case of the so-called *blind search* (completely unknown position, frequency and spin-down) the resulting parameter space is so large that the required computing power for this optimal method is completely unreasonable. In table 1 the main quantities related to the optimal search are reported. In the appendix we show how the number of points in the parameter space is calculated.

For this reason *hierarchical* methods have been developed, which are not optimal but which permit the loss of sensitivity for a given computing power to be limited (Papa *et al* 1997, Frasca 2000). They alternate coherent steps, typically based on FFTs, with incoherent ones, selecting some candidates during the incoherent step and refining the search in the following coherent phase. The incoherent steps are normally based on the Radon transform ('stack–slide' search, see Brady and Creigthon (2000)) or the Hough transform. These methods start from a time–frequency map of the detector data and produce a false alarm probability map in the space of source parameters. In each incoherent step candidates with a map value below

a given threshold are selected. The hierarchical method developed in the Virgo experiment for the detection of periodic sources is described in the following.

The starting point is the construction of a short FFT database (SFDB) containing the FFTs of the data. The 4 kHz h-reconstructed data of the Virgo detector will be used. They will be filtered and undersampled depending on the maximum frequency we want to search for. The data of this subset are divided into interlaced (by half) chunks of duration T_{FFT} each properly windowed in order to reduce the dispersion of power due to their finite length. The length of the FFTs is chosen in such a way that a signal, if present, would be completely contained in a single frequency bin. In order to optimize the sensitivity, the SFDB is divided into four blocks, one for each frequency band which will be analysed. The four frequency bands are

$$\begin{aligned} B_1 &= [500\,\mathrm{Hz}, 2\,\mathrm{kHz}], & B_2 &= [125\,\mathrm{Hz}, 500\,\mathrm{Hz}], \\ B_3 &= [31.25\,\mathrm{Hz}, 125\,\mathrm{Hz}], & B_4 &= [5\,\mathrm{Hz}, 31.25\,\mathrm{Hz}]. \end{aligned} \tag{1}$$

From the SFDB, calculating the periodograms (the square modulus of the FFT) and selecting the maxima above a given threshold, we obtain a time–frequency *peak map*. The frequency resolution is given by $\delta f = T_{\mathrm{FFT}}^{-1}$. In the absence of a signal the points in the *peak map* are distributed in a random way; if a signal is present, with high enough signal-to-noise ratio, some of these points will be distributed along the trajectory of the received frequency. The Hough transform is applied to the *peak map*. It maps each point of the *peak map* into the space of source parameters. To clarify this point, let us assume that the only parameters are the two coordinates of the source and that the source emits a signal at a frequency f_0. Moreover, we assume that at a given time t a peak at frequency $\tilde{f}$ is present in the *peak map*. The Hough transform maps this peak into the set of points, on the celestial sphere, where a source emitting a signal with frequency f_0 could be located in order to produce at the detector a peak at $\tilde{f}$. This set of points is a circle. This circle has a centre given by the direction of the detector velocity vector $\vec{v}$ at time t, calculated with respect to the solar system barycentre, and a radius ϕ which can be calculated from the equation of the Doppler effect:

$$\tilde{f} - f_0(t) = f_0(t)\frac{v(t)}{c}\cos(\phi). \tag{2}$$

As a matter of fact, this would be exact if we worked in a continuum. Due to the frequency discretization, the frequency of the peak has an indetermination given by the bin width δf and then the possible sky locations form an annulus with radii

$$\cos(\phi)_{\max} = \frac{\tilde{f} - f_0(t) + \frac{1}{2}\delta f}{f_0(t)\frac{v}{c}}, \qquad \cos(\phi)_{\min} = \frac{\tilde{f} - f_0(t) - \frac{1}{2}\delta f}{f_0(t)\frac{v}{c}}. \tag{3}$$

Summing in a suitable way the annuli corresponding to all peaks present in a given *peak map*, we have the final Hough map where candidates are selected. Their parameters are used to correct data for the next coherent step. If the time–frequency peaks were due only to a signal, the annuli would intersect in a single 'point' of the Hough map identifying the source position. Due to the presence of noise many candidates will be found. Obviously, not only the frequency space but also the celestial sphere is discretized. This means that 'points' in the Hough map are, in fact, pixels. The choice of the best kind of tiling is not trivial and is important in order to speed up the computation and increase the signal-to-noise ratio. At the moment, we have chosen a rectangular tiling, obtained with standard parallels and meridians, but other kinds of tilings are investigated. The resolution on the celestial sphere, that is the pixel dimension, is $\mathrm{res} = \left(\frac{v}{c}\frac{L_{\mathrm{FFT}}}{2}\right)^{-1}$ (see the appendix for more details). An under-resolution factor of 2 is used for the highest frequency band, in order to limit the computing needs.

Table 2. Main quantities for the full-sky incoherent step. A total observation time $T_{\rm obs} = 4$ months and a minimum neutron star decay time $\tau \equiv \frac{f_0}{\dot{f}_0} = 10^4$ years are assumed. The four frequency bands defined in equation (1) are considered. $L_{\rm FFT}$ is the FFT length (number of points), $N_{\rm tot}$ is the total number of points in the parameter space, CP is the computing power needed, calculated assuming that the computation takes place in half the total observation time, $h_{\rm min,eff}$ is the sensitivity after selection of 10^9 candidates, $F_{\rm loss}$ is the 'effective' loss factor of the hierarchical search with respect to the optimal one. In the sensitivity evaluation a detector noise power spectral density $S_h = 10^{-46}\,{\rm Hz}^{-1}$ is considered.

	$L_{\rm FFT}$	$N_{\rm tot}$	CP (Gflops)	$h_{\rm min,eff}$	$F_{\rm loss}$
$B1$	4194 304	1.90×10^{14}	603.43	1.23×10^{-25}	2.76
$B2$	4194 304	1.90×10^{14}	150.86	0.87×10^{-25}	2.09
$B3$	2097 152	1.19×10^{13}	4.71	0.73×10^{-25}	1.91
$B4$	1048 576	7.42×10^{11}	0.147	0.62×10^{-25}	1.77

In the proposed hierarchical scheme we have up to four iterations in each of which the length of FFTs is increased by a factor of 16, until a length corresponding to the total observation time is reached. At the end of the first iteration a number of candidates up to about 10^9 will be selected. This number is constrained by the necessity that the computing power needed for the subsequent analysis and the disk space to store these candidates are not too large. In the next iterations the number of candidates will rapidly reduce to very small values and so the false alarm probability becomes negligible. In table 2 the main quantities relative to the full-sky incoherent step are given. The computing power needed for the whole hierarchical procedure is about twice that needed for the first step.

In the *blind search* the first Hough transform takes into account the whole parameter space, this is why it is also called the *full-sky* incoherent step. Starting from the second step, the Hough transform is performed only considering a small region of parameter space surrounding each candidate. As a consequence, the first incoherent step is the most computationally heavy and great care must be taken in trying to implement an efficient method to calculate the Hough transform. This is the main subject of the next section.

The loss of 'nominal' sensitivity, which is the amplitude of the signal that could be detected with SNR equal to 1, of the hierarchical method with respect to the optimal one is given by the factor

$$\left(\frac{T_{\rm obs}}{T_{\rm FFT}}\right)^{1/4}. \tag{4}$$

However, in order to calculate the 'effective' sensitivity loss, the number of selected candidates at the first step must be taken into account. The Hough map distribution can be approximated as a Gaussian and the selection of 10^9 candidates corresponds to a threshold in the map number count at about 4 sigma, i.e. a further reduction of sensitivity of 2. On the other hand, in the optimal detection the reduction of sensitivity due to the selection of 10^9 candidates is greater, about a factor 6, due to the larger number of points in the parameter space. The 'effective' loss factor of the hierarchical search with respect to the optimal one is given in the last column of table 2.

3. Implementation of the Hough transform

An effort has been made in order to outline an efficient way to implement the Hough transform. Many algorithms have been studied and several codes have been written, trying to optimize

their performances. Much of this work, mainly devoted to the case of search in small 'patches' in the sky, has been done in collaboration with the GEO group at AEI (Golm, Germany) and CASPUR (Consorzio Interuniversitario per le Applicazioni di Supercalcolo per Università e Ricerca). Here we focus attention on the code developed in Rome for the *full-sky* Hough transform.

The basic problem, as we have stated before, is that of 'drawing' annuli on the celestial sphere. An annulus is delimited by two circles. Each circle can be split into two halves, a 'left' border and a 'right' border. The algorithm we have used is based on look-up tables (LUT), which has been shown to be largely more efficient than the other methods we have studied. A LUT, in our case, is a C array containing the list of borders, each with the corresponding list of its point coordinates. More precisely, given a source reference frequency f_0 and a time t_0 to which a detector velocity vector direction (λ_0, β_0) corresponds, the LUT contains the coordinates (λ, β) of the points belonging to the borders with all possible values of radius and β_0 (remember that (λ_0, β_0) defines the centre of annuli for a given t_0). In practice, several symmetries can be exploited to simplify the building of the LUT. First, the LUT does not need to contain both 'left' and 'right' borders (the latter can be obtained from the former using symmetry with respect to the meridian λ_0). Using invariance of the β coordinate of a border point with respect to shifts in λ, the same LUT can be used to build borders for all possible λ_0. That is, the same LUT can be used for all times.

On the celestial sphere we use a system of ecliptical coordinates. The detector velocity vector makes, in time, small oscillations around the ecliptic, with an opening angle of about 1° and periodicity of 1 day. This belt around the ecliptic defines the region where the centres of annuli can be found. In order to take into account the deformation of annuli passing near the poles (if we project the celestial sphere onto a plane), the position of the centres is identified using a finer grid with respect to that used for drawing annuli (i.e. the LUT is built considering a number of possible values of β_0 larger than the number of pixels contained in the allowed belt width). Using ecliptical coordinates some symmetries of the problem can be better exploited in the construction of the LUT. For instance, when building the LUT we can consider just half the width of the belt where β_0 can be found, and this holds for every coordinate system, but in ecliptical coordinates the width of this belt is minimum, thus reducing the LUT dimension and the time needed to build it.

Another property, easy to show, is that once a LUT has been built for a given reference frequency f_0, it can be used for a range of frequencies at least of the order of the Doppler band for f_0, depending on the value f_0 itself: the lower the f_0, the larger the range of validity of the LUT.

Once we have built the LUT, we use it to build the Hough map (HM). First, for each peak in a *peak map* we access the portion of LUT we need and use it to calculate the set of borders corresponding to that peak (in this operation the right ascension of points, which is stored as a continuous variable in the LUT, becomes a discrete variable). In fact, we use the peaks corresponding to ten consecutive times to build what we call a partial Hough map derivative (PHMD). It is a 'derivative' because only annuli borders are taken into account at this stage.

The memory occupation of LUTs, calculated for the maximum frequency of each of the four frequency bands we consider, is

$$M_{\mathrm{LUT}} = \begin{cases} 1.3 \text{ MB at 2 kHz} \\ 1.3 \text{ MB at 500 Hz} \\ 0.17 \text{ MB at 125 Hz} \\ 0.022 \text{ MB at 31.25 Hz.} \end{cases} \tag{5}$$

The times needed to build LUTs for the same frequencies on our reference machine, a PC equipped with the Xeon processor at 2.4 GHz, are

$$T_{\mathrm{LUT}} = \begin{cases} (51.9 \pm 0.2) \text{ ms at 2 kHz} \\ (51.9 \pm 0.2) \text{ ms at 500 Hz} \\ (6.55 \pm 0.01) \text{ ms at 125 Hz} \\ (940 \pm 2)\ \mu\text{s at 31.25 Hz.} \end{cases} \tag{6}$$

These numbers are calculated averaging over 200 iterations. The timings we obtain for the construction of PHMDs (over ten times), on our reference machine, using a simulated *peak map* built assuming Gaussian noise and a threshold for selecting peaks thr $= 2.2\sigma$, are the following:

$$T_{\mathrm{PHMD}} = \begin{cases} (32 \pm 2) \text{ ms at 2 kHz} \\ (31 \pm 2) \text{ ms at 500 Hz} \\ (5.8 \pm 1) \text{ ms at 125 Hz} \\ (0.65 \pm 0.2) \text{ ms at 31.25 Hz.} \end{cases} \tag{7}$$

These numbers, which are not particularly useful, can be converted into the number of clock cycles needed to increment by one the number count in a pixel. The results are:

$$N_{\mathrm{cycle}} = \begin{cases} (63 \pm 4) \text{ at 2 kHz} \\ (63 \pm 4) \text{ at 500 Hz} \\ (45 \pm 8) \text{ at 125 Hz} \\ (21 \pm 6) \text{ at 31.25 Hz.} \end{cases} \tag{8}$$

This is the measure of performance we use to compare different codes. At the moment, these timings are dominated by the increase of the pixels count in the PHMD, which is the main bottleneck in the building of the map (contributing up to about 80% to T_{PHMD}). We are now working to eliminate it. T_{PHMD} is the main time scale in the whole Hough transform because taking into account spin-down corresponds to performing a suitable sum of PHMDs, corresponding to different times and frequencies, which produces for each spin-down value a total Hough map derivative (HMD), and the calculation of the final Hough map (HM) consists in the integration of the total HMDs (and this operation is done just once for every spin-down value, so it is not relevant from a computational point of view). Then, it emerges that the time for building LUTs is generally small with respect to the time for building all the needed PHMDs. This can be easily seen considering that with $T_{\mathrm{obs}} = 4$ months, for instance, the number of PHMDs to be summed, for each spin-down value, is (assuming that each PHMD is built using the peaks corresponding to ten consecutive times)

$$N_{\mathrm{PHMD}} = \begin{cases} 989 \text{ at 2 kHz} \\ 247 \text{ at 500 Hz} \\ 124 \text{ at 125 Hz} \\ 62 \text{ at 31.25 Hz} \end{cases} \tag{9}$$

and then the total time to build a HM is $\approx N_{\mathrm{PHMD}} T_{\mathrm{PHMD}} \gg T_{\mathrm{LUT}}$.

We stress the fact that this kind of procedure is particularly suited to be implemented in a distributed computing environment: each node can take a portion of the parameter space, e.g. a range of frequencies, and analyse it without interacting with the other nodes.

4. Computing framework

As seen in the previous section, the hierarchical procedure though strongly cutting the non-realistic computing power needed in the optimal procedure, still requires large computing

resources. The larger the available computer power the deeper the search. We aim at reaching a total sustained computing power of the order, at least, of 1 Tflops. It is not obvious, at least at the level of current technology, that such power can be collected all in one place. Then, a distributed computer architecture is needed. The Virgo computing model is organized, concerning the off-line analysis, according to a multi-tier hierarchy (Virgo Collaboration 2000). Data are produced at the experiment site, which is the tier-0. A large fraction of the computing power for off-line analysis will be collected at Cnaf in Bologna and Lyon (tier-1), where data will also be stored. Other sites, such as Rome and Naples will act as tier-2 centres, each holding, at regime, a fraction of the order of 20% of the total available computing power. Also smaller tier-3 centres are foreseen. In this context, the emerging *grid* technologies can play a very important and useful role. They enable the coordinated and coherent use of largely distributed resources in a complete transparent way for the user who submits jobs and accesses data as if everything were located on his workstation. Concerning the Rome tier-2, we are moving towards a mixed architecture in which some machines will work as a standalone farm and others as elements of the grid environment. Among the different grid projects which have started in recent years, we have joined INFN-Grid, which is strongly connected to the European DataGrid project[1] (EDG). This EU funded project has been started mainly to answer the computing needs of future LHC experiments. It has already produced several software releases, which enable the 'gridification' of a local cluster of computers. In the following we summarize the EDG architecture; for more information see, e.g., Gagliardi *et al* (2002).

4.1. Overview of EDG

A 'grid' is a collaborative set of computing, data storage and network resources, belonging to different administrative domains, that has knowledge about the status of its components through active, distributed information services. It allows certified users belonging to multi-domain virtual organizations to access large amounts of resources via single login and manages concurrent access by a large number of dispersed users.

The EDG architecture is based on the 'grid' architecture proposed by Ian Forster and Karl Kesselmann. We can distinguish four main layers: the physical fabric layer, which deals with the basic management of computing fabrics, storage and networking (automatic installation of the operative system, installation of EDG software, etc); the basic services layer, mainly based on Globus2 toolkit, which concerns services such as authentication, secure file transfer (GridFTP), information system, etc; the grid middleware, dealing with workload management, data management and monitoring system; the application layer, which contains the experiments software. Seven different types of machines are present in EDG. The access point of the user to the grid is a machine called *User Interface* (UI), from which the user, after authentication, will submit jobs. The request is sent to the *Resource Broker* (RB) which matches the job requirements to the available resources; in doing this, the RB interacts with the *Information Service* (IS) which publishes information on the available grid computing resources, and with the *Replica Catalogue* (RC), which manages data location. A computing resource is called a *Computing Element* (CE) and is given by a gatekeeper and one or more *Working Nodes* (WN). The gatekeeper is the front-end interfacing the grid environment to the underlying local cluster. Moreover, it also hosts the server for the Local Resource Management System (LRMS), typically a batch system like PBS or LSF. The WNs are the machines where computations actually take place and host the clients of LRMS. The data are located in the *Storage Elements* (SE). In the matchmaking process done by the RB the characteristics of

[1] Website: eu-datagrid.cern.ch/eu-datagrid/.

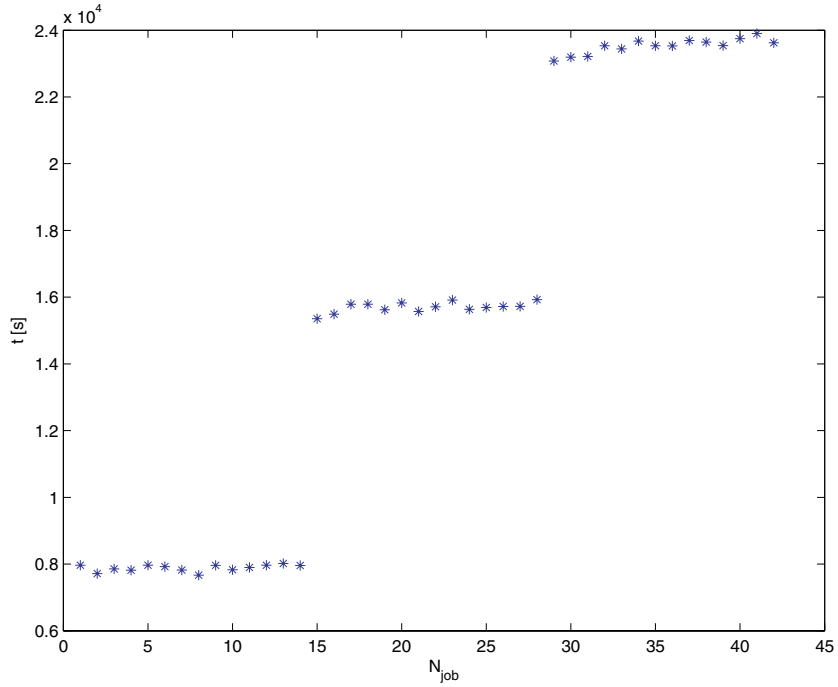

Figure 1. Timing for the grid test. The time (in seconds) needed to complete each of the submitted jobs is plotted. The number of available cpus for computation was 14. The total number of submitted jobs was 42.

the WNs (number of available processors, processor speed, memory and so on) versus job requirements and the location of the needed data are taken into account.

Using the EDG software, we have deployed a small computational grid and have performed some tests of geographically distributed computing with our codes for periodic source search. These tests will be described in the following section.

5. Description of grid tests

The aim of these tests was to understand if the use of a computational grid can be useful for our data analysis problem. In the grid tests several sites of the INFN-Grid testbed have been involved. We have used the code for the Hough transform (with no spin-down). The input data were peak maps produced starting from simulated data. The INFN-Grid RB, located at CNAF, has been used for global workload management. At the farm level, the local resource management system was PBS. The tests have been carried out according to the following scheme.

Input data have been located on the SEs. Jobs have been submitted from the UI in Rome. Each job was staged to a given WN by the RB. Also input data were copied to the local disk of that WN from the nearest SE. Once the computation was completed, output was sent back to the SE. The important point is that the grid offers independence from execution location, that is the user does not know where his job will be run, and independence from data location, i.e. the user does not know where the data his job needs are. In one of the tests we have submitted several jobs 'at the same time' and we have verified that they spread homogeneously

among the available WNs, as expected. The overhead time due to the managing activity of the RB is negligible (if the job duration is long enough; in our case about 2 h); this means that the time needed to complete N jobs is nearly equal to the time needed for just one job, T_{job}, if $N \leqslant M$, where M is the number of available cpus. If $N > M$ the total time needed is $T_{\mathrm{tot}} = T_{\mathrm{job}}\ \mathrm{ceil}\left(\frac{N}{M}\right)$. This is a consequence of the complete independence of each job from the others. This behaviour is shown in figure 1 where the timings relative to the test are plotted.

6. Conclusions

In this paper we have focused our attention on the full-sky incoherent step of the hierarchical procedure developed in the search for periodic signals in the data of the Virgo detector. This step is the most computationally heavy, because it involves the whole parameter space to be searched, and is based on the Hough transform. We have devised an efficient way to implement it, based on the use of look-up tables. Here, we have described the main aspects of the implementation and the performance measurement we have done. We have also shown that the code is very suitable to be run in a distributed computing environment, due to the fact that the whole analysis task can be divided into several smaller tasks, each independent of the others. In particular, we have tested the code in the context of emerging grid technologies which are today passing through a phase of strong acceleration. Tests have been performed among several sites, where the EDG (European DataGrid) software has been installed, and have been successful in showing that grid computing can be an important and efficient tool for our kind of analysis. In the near future our main targets are to complete and integrate all the parts of the software for the periodic source search, to test it and apply it to the analysis of the engineering data of the Virgo detector. This will be done both on local clusters and in a grid environment.

Acknowledgments

We want to anknowledge M A Papa and A Sintes for useful discussions and fruitful collaboration, G Amati and F Massaioli for their useful advice in the software development, M Serra, G Tortone and the Naples Virgo group for their support and collaboration in grid tests. Moreover, we also thank the anonymous referee for the suggestions aimed at improving the clarity of the paper.

Appendix

Here we derive the general formula for the number of points in the parameter space. This number enters the calculation of the computing power needed for the analysis. Let us indicate with T_{FFT} the duration of the FFT (equal to the total observation time, T_{obs}, for optimal detection), with f_0 the maximum frequency for which the analysis is done (half of the sampling frequency). In the hierarchical search we have different values of f_0 for the different frequency bands. The maximum length of FFT is

$$L_{\mathrm{FFT,max}} = 2T_{\mathrm{FFT,max}} f_0, \tag{10}$$

where $T_{\mathrm{FFT,max}}$ is the maximum FFT time duration such that a signal, if present, would be completely contained in a frequency bin. The length of FFT, L_{FFT} is obtained approximating $L_{\mathrm{FFT,max}}$ to the nearest power of 2. Then, the effective FFT duration is

$$T_{\mathrm{FFT}} = \frac{L_{\mathrm{FFT}}}{2 f_0}. \tag{11}$$

The number of independent frequencies is

$$N_f = BT_{\mathrm{FFT}}, \tag{12}$$

where B is the width of the frequency band. The mean number of frequency bins in the Doppler band is

$$N_{\mathrm{bin}} = \frac{v}{c}\frac{L_{\mathrm{FFT}}}{2}, \tag{13}$$

where v is the modulus of the detector velocity, and the angular resolution in the sky is its inverse, so that the total number of pixels in the sky is

$$N_{\mathrm{sky}} = 4\pi N_{\mathrm{bin}}^2. \tag{14}$$

Concerning the number of spin-down parameter values, expanding the signal frequency as a power series of the time and considering the maximum frequency variation in a time T_{obs}, at the various orders, we find the following expression for the number of spin-down parameters of order j:

$$n_{\mathrm{sd},j} = L_{\mathrm{FFT}}\left(\frac{T_{\mathrm{obs}}}{\tau}\right)^j, \tag{15}$$

where τ is the neutron star minimum decay time. Then, the total number of spin-down parameters is

$$N_{\mathrm{sd,tot}} = \prod_{j;n_{\mathrm{sd},j}\geqslant 1} n_{\mathrm{sd},j}. \tag{16}$$

Finally, the total number of points in the parameter space is

$$N_{\mathrm{tot}} = N_f N_{\mathrm{sky}} N_{\mathrm{sd,tot}}. \tag{17}$$

Using this expression we have found the numbers listed in tables 1 and 2.

References

Brady P R and Creighton T 2000 *Phys. Rev.* D **61** 082001
Frasca S 2000 *Int. J. Mod. Phys.* D **9** 369
Gagliardi F, Jones B, Reale M and Burke S 2002 *Proc. Performance 2002* p 480
Papa M A, Astone P, Frasca S and Schutz B 1997 *Proc. 2nd Workshop on Gravitational Wave Data Analysis* p 241
Virgo Collaboration 2002 *Virgo Computing Model*