Journal of Computational Interdisciplinary Sciences (2011) 2(2): 87-92

Printed version ISSN 1983-8409 / Online version ISSN 2177-8833
http://epacis.net
doi: 10.6062/jcis.2011.02.02.0035

Grid environment for turbulent dynamics in cosmology

Renata S.R. Ruiz[1], Haroldo F. Campos Velho[1], César A. Caretta[2],
Andrea S. Charão[3] and Roberto P. Souto[4]

Manuscript received on October 20, 2010 / accepted on May 26, 2011

ABSTRACT

A recent proposal, addressing the organization and evolution of large-scale structures in the Universe, suggests a similarity between cosmological evolution and the dynamics of a turbulent fluid. In this work, we present some optimization procedures for extending the analyses of turbulent signatures in the gravitational potential energy spectrum of dark matter haloes of galaxies at different redshifts. These include the development of a parallel version of the Friends-of-Friends algorithm, for identifying the dark matter haloes, and the implementation of a grid environment. Our data are from the N-body simulations run by the Virgo Consortium. The spectra for different redshifts are computed as independent tasks, therefore, we can use parallelization and a grid environment. We employed the OurGrid middleware in such grid experiment. Here we present the first results demonstrating the feasibility of implementing a computational grid for Astrophysics in Brazil. The obtained results fully corroborate the previously found turbulent-like signatures.

Keywords: computational data analysis and simulation in general sciences, grid computing, parallel friends-of-friends, turbulence and cosmology.

1 INTRODUCTION

In recent years, the volume of astronomical data has been increasing exponentially, especially from observations and simulations. This shows the necessity for efficient for efficient data storage, mining and processing. For addressing the enhancing demand of the computing resources and providing efficient tools for analysis and exploitation of the data, the astronomical community has developed the Virtual Observatory. This represents a collective effort in which several research centers, in different parts of the world, work together to provide the resources needed for a full and effective exploitation of these large and complex data sets [1]. In Brazil, for preparing the Brazilian astronomical community for this demand, a recent initiative was the development of the BraVO Project (**Bra**zilian **V**irtual **O**bservatory: http://www.lac.inpe.br/bravo/). More specifically, this project intends to generate investment in information technology, with particular emphasis on computational infrastructure, data grid, data processing, and data mining.

This work presents the first results demonstrating the feasibility of implementing a computational grid for astrophysics in Brazil. The technology of grid computing provides mechanisms to share and coordinate the use of several computational resources (supercomputers, clusters, storage devices, etc.) geographically distributed in different institutions in order to create a single virtual computer capable of achieving high processing

Correspondence to: Renata S.R. Ruiz – E-mails: renata@lac.inpe.br; rennaruiz@gmail.com
[1]Laboratório Associado de Computação e Matemática Aplicada, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brazil.
[2]Departamento de Astronomía, Universidad de Guanajuato, Guanajuato, Gto., Mexico.
[3]Departamento de Eletrônica e Computação, Universidade Federal de Santa Maria, Santa Maria, RS, Brazil.
[4]Laboratório Nacional de Computação Científica, Petrópolis, RJ, Brazil.

rate and storage power [2, 3]. Institutions that currently are part of this grid are: Instituto Nacional de Pesquisas Espaciais (INPE, Brazil), Universidade Federal de Santa Maria (UFSM, Brazil), and Universidad de Guanajuato (UG, Mexico).

The current application shown here was for determining the gravitational potential energy spectra at different redshifts (representing distinct ages of the Universe), for dark matter haloes of galaxies from a N-body simulation [4]. The standard model of cosmological structure formation is based on the gravitational Jeans instability criterion. This model ignores typical fluid effects such as viscous and turbulent forces, and diffusivity [5, 6, 7, 8, 9]. Recent studies, taking into account the simulated dark matter haloes of galaxies, indicated a possible signature of dynamical turbulence in the large scale clustering of matter [10]. These results showed that, for the galaxy haloes, the gravitational energy is organized according to the –5/3 scaling in the range from 15 to 50 $h^{-1}Mpc$. This is an interesting remark, since the turbulent cascade for energy transfer follows the –5/3 power law. This power law is known as the 1941 Kolmogorov's law [11]. Nevertheless, deeper investigations are necessary to obtain a complete picture of the dynamical characteristics of the cosmological evolution, and to address a more definitive conclusion about the turbulent behavior of this process.

For computing the gravitational energy spectrum, a preprocessing to identify dark matter haloes from the distribution of dark matter particles is necessary. A cluster analysis using the Friends-of-Friends (FoF) approach [12, 13] is usually employed for this task. FoF algorithm has a high computational cost, with complexity $\approx O(N^2)$. Since in the modern N-body simulations billions of particles are used, the performance of FoF algorithm must be improved. Here, we present a parallel version of the FoF algorithm (P-FoF), and evaluate its performance by speed-up and efficiency analyses, in a region of the cube domain of a Virgo Project simulation with 2,245,649 particles. This parallel version of the algorithm is also probed in the grid computing environment.

In order to calculate the gravitational potential energy spectra, the P-FoF was employed for identifying galaxy-sized haloes in the full region of the Virgo Intermediate Scale Simulation ($L = 239.5\ h^{-1}Mpc$, and $N = 256^3$ particles of mass $6.86 \times 10^{10} M_{\odot}$). Therefore, the main goal of this paper is to present the development and implementation of a parallel version for the Friends-of-Friends (FoF) algorithm, and also to obtain the spectrum for different redshifts as an application that can intensively use a grid environment. The software platform used for implementation of the grid was the middleware Ourgrid [14].

The sections of the paper are organized as follows. Section 2 gives a description of the FoF algorithm, the parallel strategy, and the performance analysis. Section 3 shows the performance of the grid environment in the calculation of the dark matter haloes by P-FoF algorithm, and details for obtaining of spectrum of the potential gravitational energy. Finally, Section 4 addresses some final remarks.

2 The Friends-of-Friends Algorithm

One of the most used methods for identifying structures in the Universe is the percolation or FoF algorithm [10, 12, 13]. The basic idea of this algorithm is as follows: we begin by considering a sphere with radius l (the sphere center is the gravity center of a given particle); if inside this sphere there are other particles, these particles will be considered as belonging to the same group (friends). After that, other spheres are considered, centered at each particle placed inside of the previous sphere (the friend particles), and the procedure is repeated applying the rule "any friend of my friend is also my friend". The process stops when there is no new friends that could be added to the group. In other words, the FoF algorithm puts together particles that have pair separations smaller than the linking length l. This length may be given by a fraction b of the mean interparticle separation. The values of b and l depend on the nature of the application [10]. The resulting groups are limited by a surface with constant density given by:

$$\frac{n}{\bar{n}} = \frac{2}{(4/3)\pi l^3}\frac{1}{\bar{n}} = \frac{3}{2\pi l^3}\bar{l}^3 = \frac{3}{2\pi}\frac{1}{b^3} \approx \frac{1}{2b^3} \quad (1)$$

where n is the local number density of objects (particles), $\bar{n}$ is the mean number density of particles, and $\bar{l}$ is the mean interparticle separation. The significant advantages of FoF are the simplicity, reproducibility, and the capacity to detect haloes of any shape [10].

2.1 Strategy for Parallel FoF (P-FoF)

Parallel computing is now a standard procedure to improve the performance of a computational code. Basically, the parallel processing may be understood as the simultaneous use of several processors to solve one problem. Thus, it allows a reduction of the execution time. According to Flynn's taxonomy [15] (one of the earliest classification system for computers and programs), a parallel architecture could be defined based on the numbers on concurrent (parallel) instructions and the data flow available for each architecture. In that taxonomy, there are four different models:

- **Single Instruction, Single Data (SISD):** a sequential computer, executing only one instruction for only one data flow. This model is known as "Von Newmann" architecture, examples are PC (personal computer) with one mono-core processor.
- **Single Instruction, Multiple Data (SIMD):** this is the case of a vector architecture, in which the same instruction is executed by several processors on several data flows.
- **Multiple Instruction, Single Data (MISD):** several units carry on different operations on the same data. Pipeline architectures belong to this type, but there are few examples of this architecture.
- **Multiple Instruction, Multiple Data (MIMD):** machines with several processors working asynchronously and independently, where each processor may be executing different instructions on different subsets of data. This architecture can be found for shared memory or distributed memory computers.

Distributed systems, as clusters and computational grids, are generally recognized to be MIMD architectures, either exploiting a single shared memory or a distributed memory space.

The concept of process can be used in parallel programming. Therefore, parallelism is obtained by simultaneously executing of a set of processes. The two most used standards for parallel programming are the MPI (Message-Passing Interface) library [16], extensively used for computers with distributed memory, and the fork-join model of process, whose standard is the OpenMP [17], much employed on shared memory computers.

The parallel implementation of an algorithm consists on the separation of tasks (where several tasks for solving a problem are separate on many processes), or domain decomposition (where the problem data are split among many processes). For the FoF algorithm, the best option for the parallel version is the domain decomposition, where each process executes part of the data. However, the data has strong dependency, and a random division of the domain could artificially separate particles that are gravitationally linked. Therefore, a post-processing strategy was implemented to identify, on interface regions between the subdomains, the groups with the same characteristics in two different sub-domains. A new clustering is performed on this sub-domain and the particles are then associated to their respective groups.

The FoF algorithm was parallelized using the Message Passing Interface (MPI) as follows: the master process reads the input file and the number of particles is divided into P equal-sized domains and distributed to the remaining processors. Each process compute and send the partial groups to the master process. The master process also runs FoF, writes its partial groups and receives the partial groups of the other processes. After this, the master process computes the post-processing and finally, creates the output files with the overall groups.

2.2 Performance Results for the P-FoF

The two most used measurements for the performance of a parallel code are the speed-up and the efficiency [16]. The speed-up is the ratio between the execution time spent in the sequential execution and the execution time using the parallel program. If $T_\sigma(N)$ represents the execution time of the sequential code, and $T_\pi(N, P)$ the execution time for the parallel code with P processors, the speed-up of the parallel code is defined as:

$$S(N, P) = \frac{T_\sigma(N)}{T_\pi(N, P)}\,. \tag{2}$$

In general, the relation is verified: $0 < S(N, P) \leq P$. The speed-up is linear (or ideal) when $S(N, P) = P$. If $S(N, P) > P$, the speed-up is called super-linear.

According to [16], the efficiency is a measurement of the use of processors in a parallel code, in comparison with the sequential code. It is defined as:

$$E(N, P) = \frac{S(N, P)}{P} = \frac{T_\sigma(N)}{P T_\pi(N, P)}\,. \tag{3}$$

For evaluating the performance of our P-FoF, the algorithm was applied to a cubic region from the Virgo project with 120 $h^{-1}Mpc$ of side, and 2,245,649 particles at redshift $z = 0$. The tests were run in a HP XC cluster, from the INPE High Performance Computing Center (C-PAD), São José dos Campos SP, Brazil. This machine has AMD-Opteron 2.2 GHz processors, with InfiniBand interconnection network of high speed with bandwidth of 2.5 Gbps. The machine has 112 CPUs, and the system allows the simultaneous execution up to 112 jobs.

Table 1 shows the time spent in the parallel execution of the P-FoF, the speed-up, and efficiency obtained with the data cited above. The efficiency of this P-FoF (last column in the table) is greater than 1. For the evaluation of the speed-up, the time spent with the pos-processing for grouping the subdomains was not considered.

Figure 1 compares the ideal speed-up (linear) and the speed-up obtained with our parallel version of FoF; we can see the super-linear speed-up for our P-FoF applied to the Virgo data. The performance analysis by speed-up and efficiency shows that the

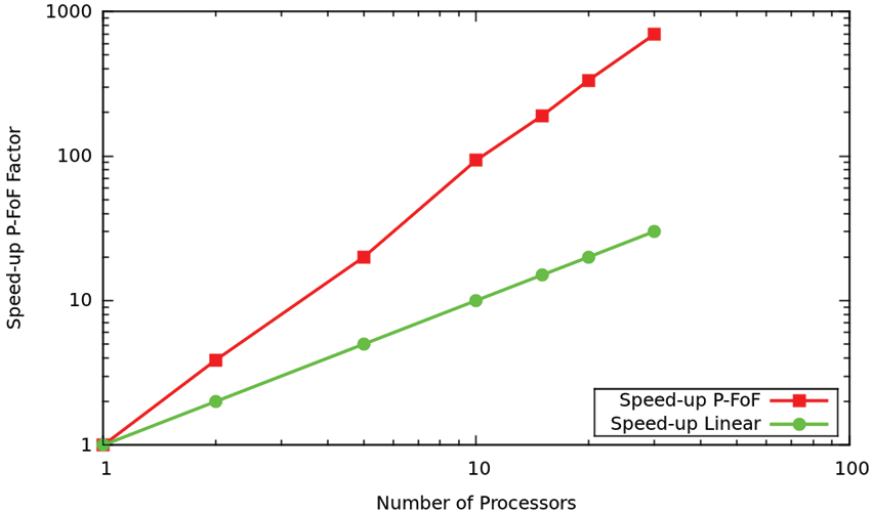

Figure 1 – Comparison between the P-FoF speed-up and the ideal speed-up.

parallel version has significantly reduced the processing time. The most important component in the CPU-time is the number of operations in the algorithm. Defining $T_{SFoF}(N) \propto N^2$ as the CPU-time for the sequential FoF, the P-FoF will be present

$$T_{PFoF}(N) \propto T_{SFoF}(N) P^{-2},$$

where P is the number of processors in the machine. The factor P^{-2} explains the super-linear behavior for the P-FoF.

Table 1 – Time, speed-up, and efficiency for the P-FoF algorithm.

# Processors	Time (min.)	Speed-up	Efficiency
1	561.15	1.00	1.00
2	145.13	3.87	1.93
5	28.00	20.04	4.01
10	6.00	93.52	9.35
15	2.96	189.58	12.64
20	1.68	334.02	16.70
30	0.81	692.78	23.09

For computing the speed-up and efficiency, the simulations were performed using a multi-processor machine for a volume of 2,245,649 particles. The sequential execution time for this volume was 9.35 hours. Taking into account the complexity of the FoF algorithm that is $\approx O(N^2)$, and considering the total volume of the Virgo Intermediate Scale Simulation ($N = 16{,}777{,}216$ particles), the total time for such sequential execution of FoF would be expected as 521 hours. Table 2 shows the time used in the parallel execution of FoF for this data volume with multi-processor computer.

Table 2 – Time for the P-FoF algorithm to volume with 16,777,216 particles.

# Processors	Time (min.)
24	51.03
48	13.65
96	4.39
144	2.50
192	1.79
240	1.45

3 Grid Computing For Turbulent Dynamics in Cosmology

The spectra for distinct redshifts are obtained as independent tasks, and hence the calculation of these spectra is treated here as a "Bag-of-Task", i.e., such tasks do not require communication during execution. This is a very important condition for using the grid environment. The P-FoF concentrates the greatest computational effort. Therefore, the analysis of the dark matter halo is evaluated as a measurement to speed-up the grid.

The grid environment was implemented using the OurGrid middleware. There are three most important components for OurGrid plataform: Mygrid, Peers, and Workers. **MyGrid** is the broker component of the OurGrid solution. MyGrid receives the application to run from the user and acts as the grid coordinator. It schedules the execution of tasks and manages all the necessary data transfers. **Peer** is the OurGrid component that organizes and provides worker machines that belong to the same administrative domain. **Worker** is the OurGrid component that allows a machine to be used for remote execution of applications.

The spectra of the gravitational potential energy were computed following a multi-step process:

- **Step 1:** Identification of dark matter haloes by using P-FoF, and indexing the particles and groups.
- **Step 2:** The strategy of domain decomposition is used by computing the properties for each halo, where each sub-domain is addressed for different processors. Properties computed: coordinates (they are obtained from the average of coordinates of particles in the halo), velocity components, velocity dispersion for the halo particles, total mass, potential and kinetic energy of the haloes. All routines used in this computation were also implemented with parallel directives employing MPI.
- **Step 3:** After calculating the properties for each halo, the boundedness criterion is applied for selecting the haloes that obey the Virial theorem. The detection of a dark matter halo by the FoF algorithm does not guarantee that the particles that compose it are gravitationally bound, since some particles may be only passing through. Thus, an additional boundedness criterion must be applied in order to take only the bounded haloes. The strategy used here picks up only the haloes that follow the Virial relation, as described in [10].
- **Step 4:** Compute the gravitational potential energy U. Following [10], the energy U is computed for each halo considering concentric spheres

$$U_j = \frac{1}{2} G m_j \sum \frac{m_i}{r_{i,j}} \tag{4}$$

where m_i and m_j are halo masses, and $r_{i,j}$ is the distance between the haloes i and j. The average energy is estimated for all haloes in each shell, and the spectra is computed for all shells.

The results for a cube with sides of 239.5 $h^{-1}Mpc$ are shown in Figura 2. The figure displays the gravitational potential energy of galaxy dark matter haloes as a function of a "wave-number" (defined from the distances between haloes). The solid line in the Figura 2 represents the straight line with slope $-5/3$. The slope –5/3 is close to the slopes presented by galaxy haloes in all the studied redshifts for wave numbers between 0.01 and 0.06 (from 17 up to 100 $h^{-1}Mpc$). Our parallel version in grid computing has fully reproduced previous results [10].

Our grid experiment consisted of 9 jobs, for obtaining the spectrum for different redshifts. The total time (T_S) is defined as the overall run-time of jobs submitted to grid nodes and corresponds to a sequential execution of a job set. The time of grid usage required to perform the whole set of jobs is defined as a

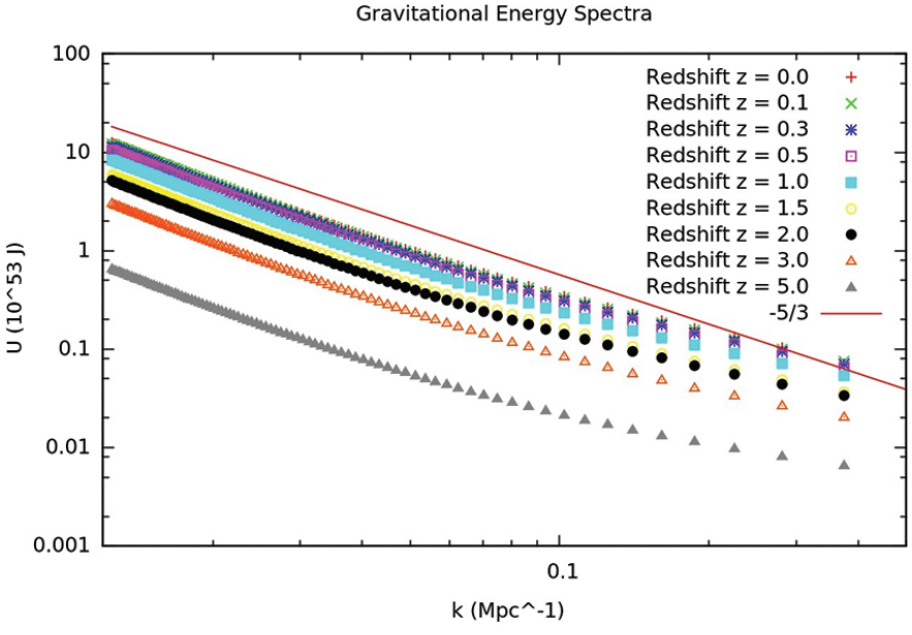

Figure 2 – Gravitational potential energy spectra for galaxy haloes in the range 2 to 100 $h^{-1}Mpc$. The distinct redshifts considered are represented by different symbols. A reference –5/3 slope solid line is shown.

grid elapsed time (T_G). The ratio T_S/T_G gives the speedup for the grid execution.

The Table 3 show that $T_S = 40.43$ and $T_G = 14.53$, then the speed-up for the grid was:

$$S = \frac{T_S}{T_G} = \frac{40.43}{14.53} = 2.78 \quad (5)$$

Table 3 – Accumulated cluster time (hours).

Cluster	Jobs	T_S (hh:mm)	T_S/Jobs (hh:mm)
C-PAD/INPE	5	14:32	02:54
LAC/INPE	3	13:57	04:39
UFSM	1	11:57	11:57
Total	9	40:26	

4 Conclusions

This paper shows the first application of the computational grid for Astrophysics in the context of the BraVO project, where the turbulent-like patterns in the large scale structure evolution are calculated from the gravitational potential energy spectra.

We developed a parallel version of the FoF algorithm (P-FoF) using a domain decomposition approach to split the data box and distribute in separate processes, followed by a post-processing in order to correct the ungrouping in the domain interfaces. This new tool will be available to the astronomical community in the BraVO project. The P-FoF was important to promote a cluster analysis as a pre-processing procedure to compute the spectrum of the gravitational energy of the dark matter haloes of galaxies. We also implemented a grid environment, using the OurGrid middleware, for distributing tasks between three institutions: INPE and UFSM in Brazil, and UG in Mexico.

By distributing the jobs along the grid, we have effectively reduced the time needed to obtain the results. The spectra of the gravitational potential energy for a cube with sides 239.5 h^{-1} Mpc follow the –5/3 scaling in the range 0.01 and 0.06 (17 to 100 $h^{-1}Mpc$), confirming the previous results that they resemble the –5/3 scaling of the turbulent cascade for energy transfer.

ACKNOWLEDGMENTS

The authors would like to thank the Virgo Consortium for supplying the data. Author R.S.R. Ruiz acknowledges FAPESP (Process: 2007/54133-0) and CAPES (Process: 4541/08-1) for the Sc.D. fellowships.

REFERENCES

[1] DE CARVALHO RR et al. 2009. The Brazilian Virtual Observatory – A New Paradigm for Astronomy. Journal of Computational Interdisciplinary Sciences 1(2): 1–23.

[2] FOSTER I. 2003. The grid: A new infrastructure for 21st century science. Physics Today, 55: 51–63.

[3] FOSTER I & KESSELMAN C. 2004. The grid: Blueprint for a new computing infrastructure. Morgan Kaufmann, San Francisco, USA, 2nd edition.

[4] JENKINS A et al. 1998. Evolution of Structure in Cold Dark Matter Universe. The Astrophysical Journal, 499: 20–40.

[5] SHANDARIN SF & ZEL'DOVICH Y. 1989. The large-scale of the universe: Turbulence, intermittency, structures in a self-gravitating medium. Reviews of Modern Physics, 61: 185–220.

[6] MADSEN M. 1996. The dynamic Cosmos – Exploring the physical evolution of the Universe. Chapman & Hall, New York, 1st edition.

[7] LINDER EV. 1997. First Principles of Cosmology, Addison-Wesley.

[8] TATEKAWA T. 2005. Lagrangian perturbation theory in Newtonian cosmology, arXiv:astro-ph/0412025v4.

[9] HAWLEY JF & HOLCOMB KA. 2005. Foundations of Modern Cosmology. Oxford University Press, Oxford New York, 2nd edition.

[10] CARETTA CA et al. 2008. Evidence of turbulence-like universality on the formation of galaxy-sized dark matter haloes. Astronomy and Astrophysics, 487: 445–451.

[11] FRISH U. 1995. Turbulence: The legacy of A.N. Kolmogorov. Cambridge University Press, New York.

[12] EINASTO J, KLYPIN AA, SAAR E & SHANDARIN SF. 1984. Structure of Superclusters and Supercluster Formation – III. Quantitative Study of the Local Supercluster Mon. Not. R. Astr. Soc., 206: 529-558.

[13] HUCHRA JP & GELLER MJ. 1982. Groups of galaxies I. Nearby groups. The Astrophysical Journal, 257: 423–437.

[14] ANDRADE WN et al. 2003. Ourgrid: An approach to easily assemble grids with equitable resource sharing. In Proceedings of the 9th Workshop on Job Scheduling Strategies for Parallel Processing, Seattle, WA, USA.

[15] FLYNN MJ. 1966. Very High-Speed Computing Systems. Proceedings of the IEEE, 54: 1901–1909.

[16] PACHECO PS. 1997. Parallel Programming with MPI. Morgan Kaufmann Publishers, Inc., San Francisco.

[17] CHAPMAN B, JOST G & DER PAS RV. 2007. Using OpenMP: Portable Shared Memory Parallel Programming. The MIT Press, EUA.