Volume Title
*ASP Conference Series, Vol. **Volume Number***
Author

Application of GPUs for the calculation of two point correlation functions in cosmology

Rafael Ponce[1], Miguel Cárdenas-Montes[1], Juan José Rodríguez-Vázquez[1], Eusebio Sánchez[1] and Ignacio Sevilla[1]

[1]*Centro de Investigaciones Energéticas Medioambientales y Tecnológicas, Av.Complutense 40, 28040 Madrid, Spain*

Abstract. In this work, we have explored the advantages and drawbacks of using GPUs instead of CPUs in the calculation of a standard 2-point correlation function algorithm, which is useful for the analysis of Large Scale Structure of galaxies. Taking into account the huge volume of data foreseen in upcoming surveys, our main goal has been to accelerate significantly the analysis codes.We find that GPUs offer a 100-fold increase in speed with respect to a single CPU without a significant deviation in the results. For comparison's sake, an MPI version was developed as well. Some issues, like code implementation, which arise from using this option are discussed.

1. Introduction

The two-point correlation function (2pcf) is a simple statistic that quantifies the clustering of a given distribution of objects. In studies of the Large Scale Structure (LSS) of the Universe, this is an important tool containing information about the matter clustering and the Universe evolution at different cosmological epochs, Peebles (1980). A classical application of this statistic is the galaxy-galaxy correlation function to find constraints on the matter density parameter Ω_m, Hawkins et al. (2003), or the location of the baryonic acoustic oscillation peak, Sánchez et al. (2011). Other examples include cross-correlation of background galaxies with the shear of objects caused by the gravitational effect on light (weak lensing), Dodelson et al. (2008).

The 2pcf measures the excess probability of finding a couple of galaxies separated by spatial distance r or angular distance θ with respect to the probability of finding a couple of galaxies separated by the same distance or angle in a random and uniform distribution. In this work we have used the angular version of the correlation function $w(\theta)$ though results are extendible to the 3-dimensional variant as well.

Landy & Szalay, Landy & Szalay (1993), found an estimator with minimum variance which is the standard one used in cosmological analyses:

$$\omega(\theta) = 1 \; + \; (\frac{N_{random}}{N_{real}})2 \cdot \frac{DD(\theta)}{RR(\theta)} \; - \; 2 \cdot (\frac{N_{random}}{N_{real}}) \cdot \frac{DR(\theta)}{RR(\theta)} \tag{1}$$

where N_{gal} is the number of galaxies in a real catalog, N_{rd} is the number of galaxies in a random catalog, $DD(\theta)$ is the number of pairs separated by an angular distance θ in the real catalog, $RR(\theta)$ is the number of pairs separated by an angular distance θ in the random catalog and $DR(\theta)$ is the number of pairs separated by an angular distance θ in the real catalog with respect to the random catalog.

2. Computational problem and previous work

The calculation of 2pcf, Eq.1, is very costly computationally so alternative strategies have been designed to approach the problem (pixelization of the map, Eriksen et al. (2004), k-trees, Moore et al. (2000)), usually at the cost of some loss of information.

Alternatively, in Roeh et al. (2009), this problem has been treated with GPUs using a different strategy in terms of shared memory usage. In particular, the authors of Roeh et al. (2009) have used a 'chessboard' strategy where arrays are passed to the global memory. This has the disadvantage of having restrictions in the input sample. Also, the particular implementation in Roeh et al. (2009) obtained results in $cos\theta$ space, thus complicating the cosmological interpretation of the result.

3. Implementation and hardware

We have implemented in CUDA the Landy-Szalay estimator with the following key features:

- Usage of shared memory (instead of global memory) for the dot product and arc-cosine operations necessary to extract the angle between two objects.
- Application of atomic operations in shared memory to make use efficiently of multi-threading when filling up the histograms (DD, DR and RR in Eq. 1). Partial histograms are generated in parallel in shared memory and later combined in a single histogram, in global memory.
- In one of the architectures we had available, we applied a multi-GPU solution using 3 GPUs, one for each of the histograms, in which DD and RR where used in one of the boards containing 2 GPUs and DR in the other for maximum efficiency.

A full description of the algorithm and its implementation can be found in Cárdenas-Montes et al. (2011). The hardware we have used to test our codes is in Table 1.

CPU	GPU	MPI
CPU with two Intel Xeon E5520 processors at 2.27 GHz	GTX295 C1060 (Tesla) C2050 (Tesla)	1920 cores (two Intel Xeon E5570 at 2.93 GHz, per node)

Table 1. Hardware specifications that we have used.

4. Results and analysis

The galaxy catalogs used are publicly available from the MICE project, Fosalba et al. (2008); Crocce et al. (2010).

In Table 2 we present a comparison between the execution time of CPU implementation and the execution time of GPU implementation.

In Fig. 1(a) we show, for MICE catalog, one of the correlation functions calculated using this code, versus the same calculation using a standard implementation in C for

Input file lines	CPU (s)	GTX295 (s)	C1060 (s)	C2050 (s)
$0.43 \cdot 10^6$	$3.60 \cdot 10^4$	$3.01 \cdot 10^2$	$2.91 \cdot 10^2$	$2.19 \cdot 10^2$
$0.86 \cdot 10^6$	$1.44 \cdot 10^5$	$1.20 \cdot 10^3$	$1.16 \cdot 10^3$	$8.76 \cdot 10^2$
$1.00 \cdot 10^6$	$1.98 \cdot 10^5$	$1.61 \cdot 10^3$	$1.56 \cdot 10^3$	$1.17 \cdot 10^3$
$1.29 \cdot 10^6$	$3.24 \cdot 10^5$	$2.68 \cdot 10^3$	$2.59 \cdot 10^3$	$1.97 \cdot 10^3$
$1.72 \cdot 10^6$	$5.76 \cdot 10^5$	———	$4.64 \cdot 10^3$	$3.51 \cdot 10^3$
$3.45 \cdot 10^6$	$2.32 \cdot 10^6$	———	$1.88 \cdot 10^4$	$1.41 \cdot 10^4$
$6.89 \cdot 10^6$	$9.22 \cdot 10^6$	———	$7.45 \cdot 10^4$	$5.61 \cdot 10^4$

Table 2. Comparison between CPU execution time and diverse GPUs execution time.

CPUs, for reference. The residuals at each point are plotted in Fig. 1(b) and are far below the expected errors due to cosmic variance, i.e., the statistical errors due to the small number of 'fields' available in the sky.

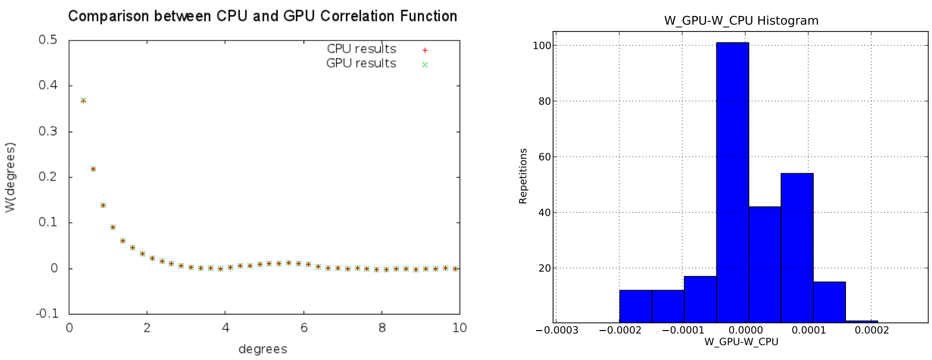

Figure 1. Panel (a, left) shows a comparison between correlation functions, the red one was calculated with the CPU code, the green one with the GPU code, while panel (b, right) shows the residuals between GPU and CPU codes. These residuals are really small and fall into the statistical errors.

We have also done a comparison between GPUs and MPI. In Fig. 2 we have our MPI time with GPUs time like a boxplot graphic.

5. Conclusions

We have developed an implementation of the Landy-Szalay two-point correlation function in CUDA to make use of the power GPUs have to offer in terms of parallelization. The speed-up with respect to a CPU is 164-fold (C2050) using the same algorithm. With respect to an implementation of k-trees in CPUs we obtain an increase of 44-fold. Several MPI configurations have been explored being the GPU implementation surpassed by the usage of more than 64 nodes, see Fig. 2.

Some options to be explored remain, such as full-blown multi-GPU implementation, coding the k-trees or extending the work to higher order correlation functions, for

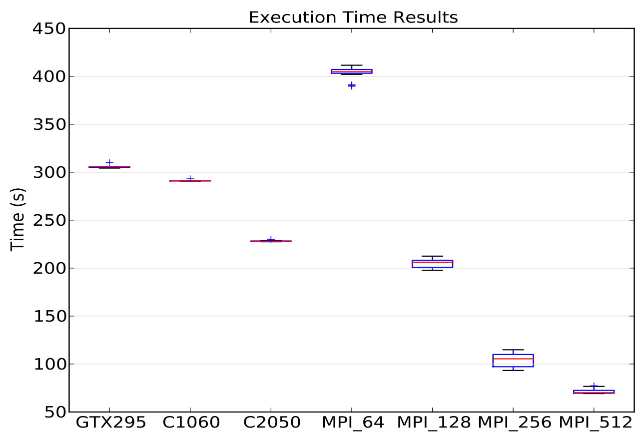

Figure 2. GPU and MPI execution time results.

other types of cosmological analyses such as understanding non-Gaussianities in the primordial perturbations.

Acknowledgments. We would like to the thank ASP for the chance to present at ADASS and we also acknowledge the use of data from the MICE simulations, publicly available at http://www.ice.cat/mice.

References

Cárdenas-Montes, M., Vega-Rodríguez, M. A., Ponce, R., Gómez-Iglesias, A., Sevilla, I., Rodríguez-Vázquez, J. J., Alvaro, E. S., & Arriero, N. C. 2011, Performance study for parallel implementations of cosmological data analysis. unpublished. Under elaboration

Crocce, M., Fosalba, P., Castander, F., & Gaztañaga, E. 2010, Mon.Not.Roy.Astron.Soc., 403, 1353. 0907.0019

Dodelson, S., Schmidt, F., & Vallinotto, A. 2008, Phys. Rev. D, 78, 043508. URL http://link.aps.org/doi/10.1103/PhysRevD.78.043508

Eriksen, H. K., Lilje, P. B., Banday, A. J., & Grski, K. M. 2004, The Astrophysical Journal Supplement Series, 151, 1. URL http://stacks.iop.org/0067-0049/151/i=1/a=1

Fosalba, P., Gaztañaga, E., Castander, F., & Manera, M. 2008, Mon.Not.Roy.Astron.Soc., 391, 435. 0711.1540

Hawkins, E., Maddox, S., Cole, S., Lahav, O., Madgwick, D. S., et al. 2003, Monthly Notices of the Royal Astronomical Society, 346, 78. URL http://dx.doi.org/10.1046/j.1365-2966.2003.07063.x

Landy, S. D., & Szalay, A. S. 1993, American Journal of Physics, 412, 64

Moore, A., Connolly, A., Genovese, C., et al. 2000, astroph0012333. astro-ph/0012333

Peebles, P. J. E. 1980, Large-Scale Structure of the Universe, Princeton Series in Physics (Sam B. Treiman)

Roeh, D. W., Kindratenko, V. V., & Brunner, R. J. 2009, in Proceedings of 2nd Workshop on General Purpose Processing on Graphics Processing Units (New York, NY, USA: ACM), GPGPU-2, 1

Sánchez, E., Carnero, A., García-Bellido, J., Gaztañaga, E., de Simoni, F., Crocce, M., Cabré, A., Fosalba, P., & Alonso, D. 2011, MNRAS, 411, 277. 1006.3226