Landy-Szalay Estimator

Nicolas Kamennoff

2013 - 2014

Introduction

Computational resources are genuinely changing from the early 2000's. This compulsion to ever growing data sets, clock speed limit and power efficiency lead new hardware toward parallelisation. From high frequency single processors, new technologies favors parallel lower frequency cores and specific hardware. This new deal comes with new challenges and requests to handle this new computational environment.

Introduced by Jim Gray [4], Data Intensive Scientific Computation (DISC) [3] represents pioneer in the field of Big Data techniques. Indeed, helped by new, ever more precise tools, the amount of data in astrophysics, genetics, meteorology and many other science growth every seconds. The DISC is even at the edge of exascale[1] which is currently out of reach for computer science.

Aside from hardware and network technologies, it relies on developers to think and to create new solutions to use these new tools that can handle new resources and that are design to optimize processing. During this first project on the Big Data courses, you will be ask to create a scalable computation tool for scientific computation.

1 The algorithm

The Landy Szalay Estimator [9] is a two point correlation function applied on extragalactic astronomical data. It computes statistical correlation score for the clustering of galaxies. Among various two point correlation estimators, this one had been developed to reduce the bias and variance of the correlation function. You can find more information in the article cited above.

1.1 Two point correlation

The correlation function usually compares two values ;

[1] Exascale is a thousand (1000) petascale, i.e. 10^{18} operation by second capable system

- The number of pairs in a user given set of data
- The number of pairs in a randomly generated set of data[2]

Pairs ?

Pairs are qualified by two objects that are at a given angular distance 'd'. For example, on the picture below (1) you can see a small example catalog. All white points are galaxies (only about 100 here).

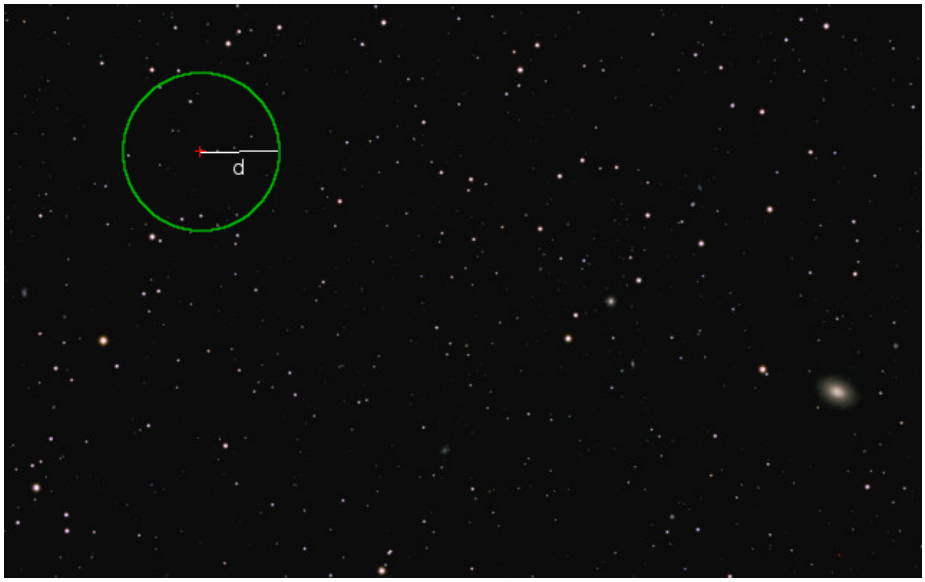

Figure 1: Example of 2 point correlation

Given the an object (in red at the center of the circle), we create a circle of radius 'd' (here in green). There is one pair each time an other object is on the green line (not inside the circle). As the line do not have natural width, we also give one called δ (delta).

As you can see on figure 2, this creates a donut's shape and every point between the two circle will be detected as forming a pair with the origin object. These points are here highlighted by red circles. So we here get 7 pairs for this peculiar object. Now, we compute the number of pairs for all points and we do the same for the random calalog.
Let NN be the number of pairs counted in the given catalog and RR the number of pairs counted in the random catalog. Then, the correlation function is :

[2]Note that the random data set must have the same number of entry than the one that is studied.

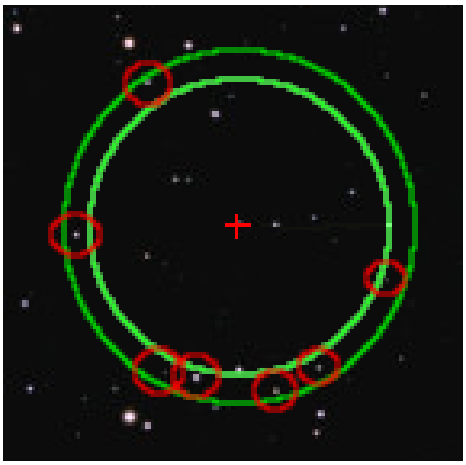

Figure 2: Zoom on object detection

$$w(\theta) = \frac{NN}{RR}$$

1.2 Landy-Szalay Estimator

Using a random data set introduces bias and emphasize variance among trials. Indeed as the number of pairs in a random data set can change, results may vary and add noise to the study. In order to limit bias and variance, Stephen Landy and Alexander Szalay have defined a new methodology. This new methodology consists in :

1. Introducing a hybrid catalog merging orignial and random data sets.
2. Using a hundred (100) different random data sets.

Thus we obtain the pseudo-algorithm as follow (figure 3):

1.3 Coordinates system

As you may have notice, we are talking about angular distance and not just mere euclidian distance. Indeed, as the sky is observed from earth in every directions, so it has to be represented as a sphere. Thus coordinates are expressed using the equatorial coordinate system (figure 4). Coordinates in this system are expressed by two different values called Right Ascension (RA) and Declination (DEC). RA represents the angular distance along the equator and then goes from 0 to 360. DEC represents the angular distance perpendicular the equator and then goes from -90 to 90.
In order to makes everything simpler, each point is considered at a distance of

Let NDS = user given data set
Set NN = number of pairs in NDS
For 100 times do
    Generate RDS = randomly generated data set with size(NDS) points
    Set RR += number of pairs in RDS
    Generate HDS = RDS + NDS
    Set NR += number of pairs in HDS
RR = $\frac{RR}{100}$
NR = $\frac{RR}{100}$
return $\frac{NN-2NR+RR}{RR}$

Figure 3: Landy-Szalay Estimator Pseudo Code

1 from the origin.

Thus angular distance **d between to objects with coordinates ra1, dec1 and ra2, dec2 can be computed by the following equation (figure ??). Note that a faster estimation exists when the area studied is smaller [13].**

1.4 Hierarchical Triangular Mesh - HTM

We will never repeat enough that knowledge representation is a burning issue for all computer science problems. While we are turning to huge data sets it becomes mandatory ! As we have to find objets at a given distance we have several choice :

- **Use chained lists and check distance between every objects (sic)**
- **Use a tree based representation and do it in a more clever way**
- **Use something else at least as clever as the previous choice**

But turning a 3D sphere in a optimized tree-based representation may be considered by some of you as tricky. Here we presents a kind of peculiar quad-tree : the Hierarchical Triangular Mesh [18]. The principle is to define an isosceles triangle based quad-tree as shown on figure 6 below.

Starting with an octahedron, the spatial shape become a sphere as shown below 7.
All details about this knowledge representation technique is available in the article from Alexander Szalay [18].

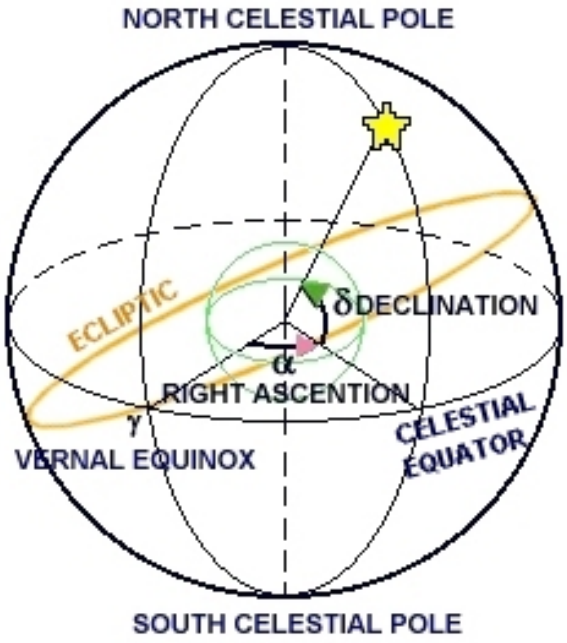

Figure 4: Equatorial Coordinate System

$$cos(d) = sin(dec1)sin(dec2) + cos(dec1)cos(dec2)cos(ra1 - ra2)$$

Figure 5: Angular Distance Equation

2 Work

You have now surely understood that this computation can take long to compute on large data sets. Indeed, and actually data sets can go from thousands to more than a billion objects. Thus, using all available resources and making optimization is no more optional.

We are now expecting you to design and code a scalable software that compute the Landy-Szalay Estimator for a given data set. You can use all available resources on your computer from CPU to GPGPU and even, bonus, distribute the solution on many machines.

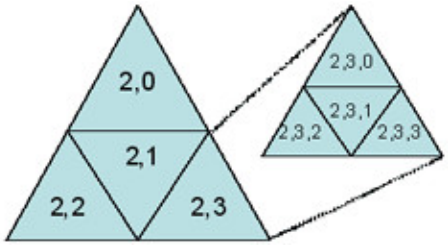

Figure 6: Trixels : nodes of an isosceles triangle based quad-tree

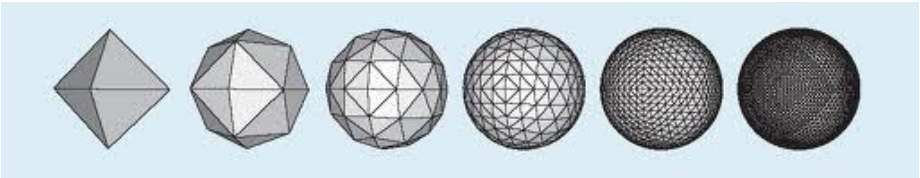

Figure 7: From octahedron to Sphere

2.1 File Format

The format of the file is quite easy to handle. Data sets will be given as CSV containing 3 piece of information : object identifier and coordinates in the ra/dec format. The object id is an integer while ra and dec coordinates are doubles. A small example is given by figure 8.

2.2 Constraints

Your program must...

- **... be scalable on CPU or GPGPU**
- **... focus on time consumption**
- **... be able to handle large data sets (> your RAM size)**

1237651758284998175,270.057301229368,0.0120026374546346
1237651758284997018,269.991933551139,0.058223848525456
1237651758284935423,269.945028331138,0.0211372217264966
1237651758284999954,270.04874283861,0.0331067068505614
1237651758284997931,270.030389675456,0.0505055672959716
1237651758285001177,270.038993108416,0.0442239906891489

Figure 8: File Example

- **... must be packaged for fedora (.rpm)**
- **... must be called M-BD1-<leader login>**
- **... must have the following synposis : $ M-BD1-<login> <dataset.csv>**
- **... write the output score on in a file <login.txt> in the home directory**

2.3 Test & Defense

This introduction module will be validated in two steps :

First, a twenty (20) minutes oral defense where you must come with a powerpoint (or equivalent) presentation and where you will be asked to show :

- **The first design of your solution**
- **Evolution of your design driven by observations**
- **Graphs showing the results you got**
- **A critic analysis of your work**
- **Leads for further optimization**

Note that this is not a formal presentation summary, feel free to adapt it.

Before the presentation you will be asked to leave your fedora package on a hard drive, so that benchmarks will be run to determine a ranking among the different projects.

To get the module, you must get at least : a well bug-free program that computes in a reasonable amount of time and a good presentation of your work.

3 General information

This project is for groups from two (2) to four (4) students from tech4 and tech5 (mixted groups are allowed).

Package name : M-BD-<login>.rpm

Groups that creates GPGPU implementation instead of CPU must warn us by mail at : acsel@epitech.eu
With subject "[M-BD1] GPGPU Landy-Szalay".

References

[1] **Chilimbi & al. Cache-conscious structure definition. *ACM SIGPLAN*, 1999.**

[2] **C. Breshears. *The art of concurrency*. O'Reilly, 2009.**

[3] **Collective. *The Fourth Paradigm - Data-Intensive Scientific Discovery*. Microsoft, 2009.**

[4] **J. Gray. Jim gray's website.** `http://research.microsoft.com/en-us/um/people/gray/`, **2007.**

[5] **Khronos Group. Opencl.** `http://www.khronos.org/opencl/`.

[6] **N. Herlihy, M. & Shavit. *The art of multiprocessor programming*. Morgan Kaufmann, 2008.**

[7] **Intel. Intel's threading building block.** `https://www.threadingbuildingblocks.org/`.

[8] **W.W. Kirk, D.B. & Hwu. *Programming Massively Parallel Processors*. Morgan Kaufmann, 2010.**

[9] **A.S. Landy, S.D. & Szalay. Bias and variance of angular correlation functions. *The Astrophysical Journal*, 1993.**

[10] **S. Lewin-Berlin. What the $#@! is parallelism, anyhow ?** `http://software.intel.com/en-us/articles/what-the-is-parallelism-anyhow-1/`.

[11] **M. Maged. Maged michael : Selected publications.** `http://www.research.ibm.com/people/m/michael/pubs.htm`.

[12] **T. & al. Mattson. *Patterns for Parallel Programming*. Addison Wesley, 2005.**

[13] **J.C. Mihos. Astronomical coordinates.** `http://burro.cwru.edu/Academics/Astr306/Coords/coords.html`.

[14] **NVIDIA. Cuda.** `http://www.nvidia.com/object/cuda_home_new.html`.

[15] **OpenMP. Openmp website.** `http://openmp.org/wp/`.

[16] **O. Pironneau. Optimisation des performaces et parallélisme en c/c++.** `http://www.ann.jussieu.fr/pironneau/calculParallele.pdf`.

[17] **Y. Shi. Reevaluating amdahl's law and gustafson's law.** `http://spartan.cis.temple.edu/shi/public_html/docs/amdahl/amdahl.html`.

[18] A.S. & al. Szalay. Indexing the sphere. *Microsoft Tech Report*, 2005.