Using GPUs for LSS analysis

Miguel Cárdenas-Montes*, Juan José Rodríguez-Vázquez, Rafael Ponce*, Eusebio Sánchez, *Ignacio Sevilla-Noarbe*
CIEMAT (Madrid)

Speeding up (your analysis) with a graphics card

Summary / Short version of the talk

- **GPUs** are **inexpensive** and **powerful** tools for **2-point correlation function** computation.

- The **code** to profit from this is **available**:

  http://wwwae.ciemat.es/cosmo/gp2pcf

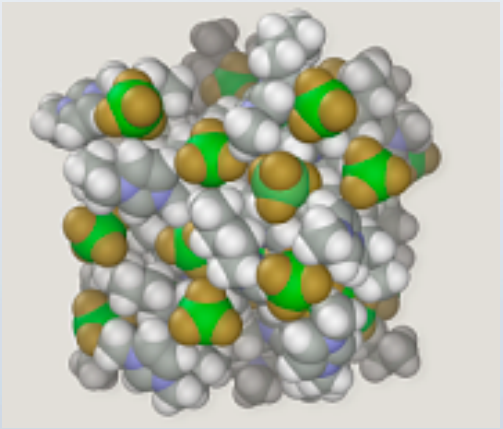
GPUs have been used for a long time

Graphics **P**rocessing **U**nit:

Originally: 2D-3D rendering (1980's - today)

Floating point operations, high parallelization

Now also physics simulations, **scientific computing** (GPGPU)



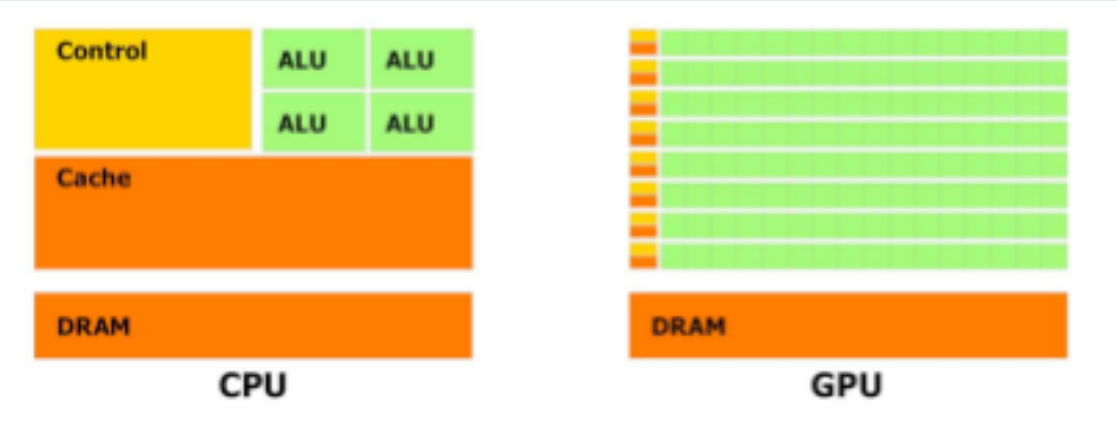
Ciemat

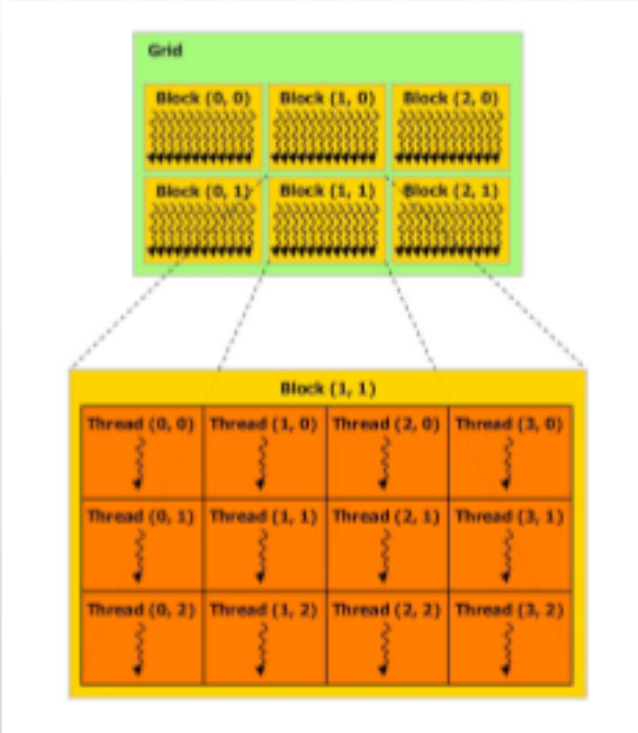
GPUs are designed specifically for parallel processing

Nowadays usually on separate card.

Most of the transistors devoted to operations.

Several blocks are sent, shared between the GPU's processors; each block: hundreds of threads doing parallel operations.





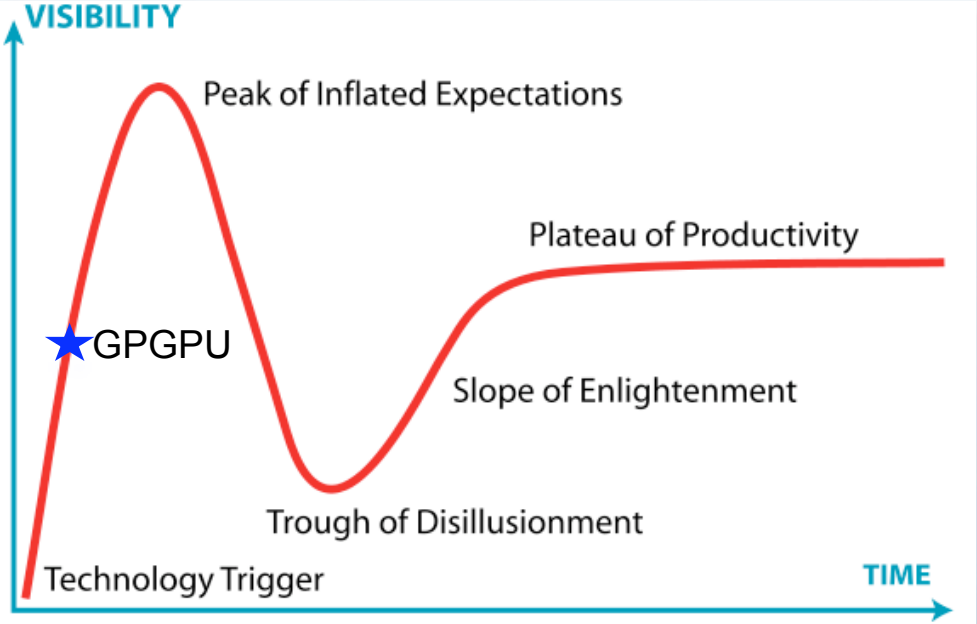
Credit: T. Jensen, H.Rasmussen, F. Rosé

Ciemat

Only very recently they are starting to be used beyond graphics processing

Multiple applications beyond graphics (GPGPU)

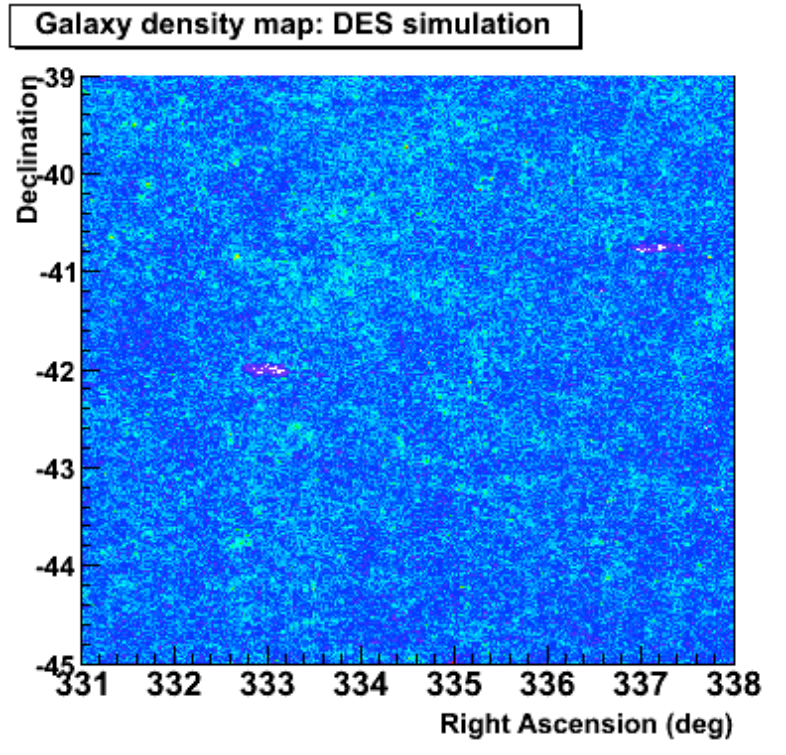
Only very specific applications (massive, parallel)



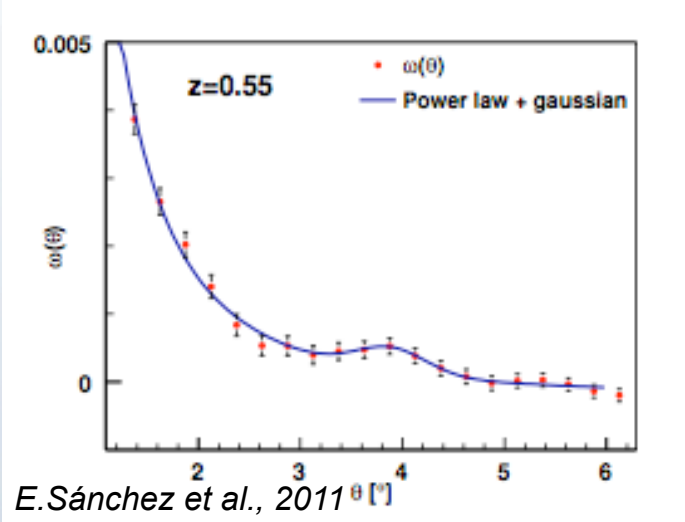
Gartner hype plot, adapted

Ciemat

Cosmology can benefit from this device: e.g. the angular 2pc function computation



Measure **deviation** of distribution from **random**. Shape, normalization, features -> **cosmological interpretation**.



E.Sánchez et al., 2011

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

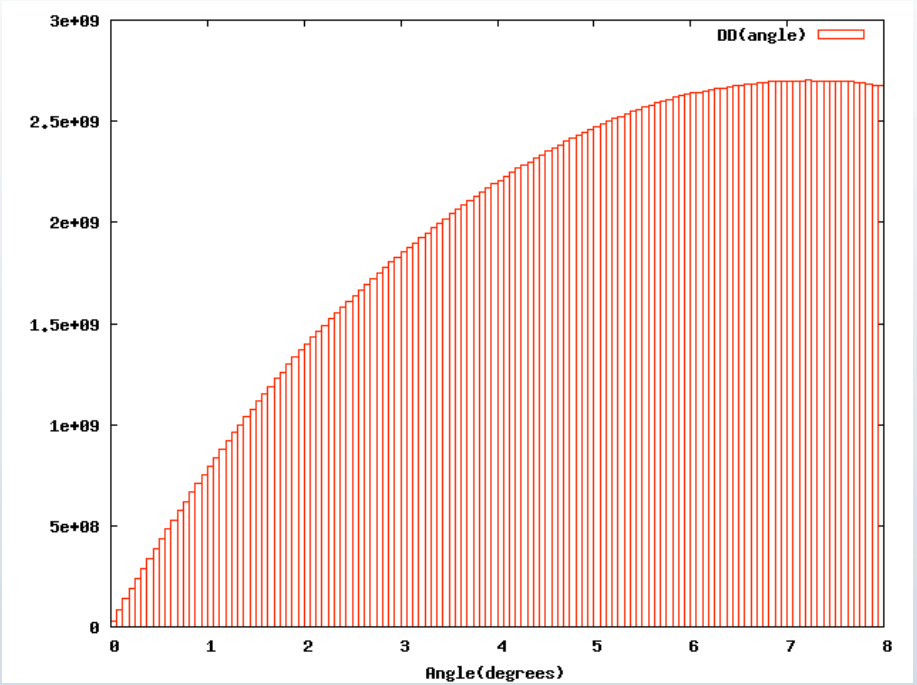
Landy & Szalay, 1993

Ciemat

We have implemented the full angular 2pc function algorithm in CUDA

CUDA is a C extension to interact with GPU using C-like syntax.

- Kernel: C-like function (*Compute angle between two galaxies*)
- *N* threads: execute kernel N times in parallel.
- Blocks: threads grouped into blocks (*DD,DR,RR histos built in each block*).



http://wwwae.ciemat.es/cosmo/gp2pcf

Ciemat

Designing the kernel needs to specially consider the features of the hardware

CPU Code

```
void binning(float *xd, float *yd, float *zd,
     int *ZZ, int numberoflines){

float angle;
int dibin[angleposition];

for (int i=0; i< numberoflines; i++){
  for (int j=0; j< numberoflines; j++){
    angle = xd[i]*xd[j] + yd[i]*yd[j] + zd[i]*zd[j];
    if(angle>1.) angle=1.;
    angle = acosf(angle) * 180./pi;
    position = int((angle)*binsperdegree);
    if(dibin[position]<binsperdegree*totaldegrees);
       dibin[position] = dibin[position]+1;
    }
  }
}
```

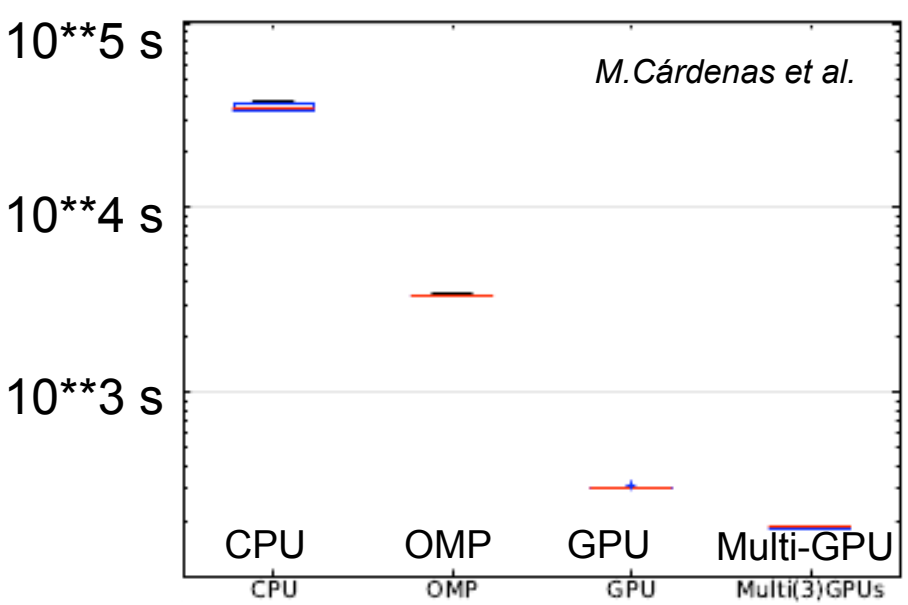
GPU Code

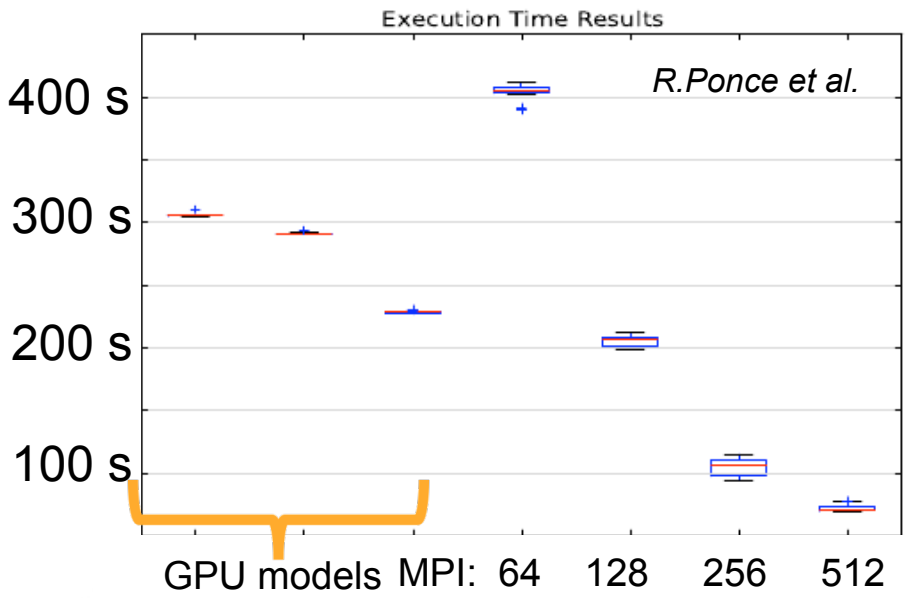
```
__global__ void binning(float *xd, float *yd, float *zd,
int *ZZ, int numberoflines){
__shared__ float angle[threadsperblock];
__shared__ int dibin[threadsperblock];
__shared__ int temp[threadsperblock];
int cacheIndex = threadIdx.x;
temp[cacheIndex]=0;
for (int i=0; i< numberoflines; i++){
 int dim_idx = blockIdx.x * blockDim.x + threadIdx.x;
 while (dim_idx < numberoflines){
   angle[cacheIndex] = xd[i]*xd[dim_idx] + yd[i]*yd[dim_idx] +
                       zd[i]*zd[dim_idx];
   __syncthreads();
   if(angle[cacheIndex]>1.) angle[cacheIndex]=1.;
   .
   .
   .
  dim_idx += blockDim.x * gridDim.x; } }
 atomicAdd( &ZZ[threadIdx.x] , temp[threadIdx.x]); }
```

A for-loop is substituted by parallel computation

Ciemat

Results: single GPU is equivalent to ~100 processors





CPU: Intel Xeon 4 cores @ 2.27 GHz
OMP and MPI: Supercomputer with 240 nodes x 8 cores @ 2.93 GHz
GPU: 3 different models
Multi-GPU: 3 x GTX295 model dividing the work of DD, DR, RR

Ciemat

Results: a million galaxies in 20 minutes

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

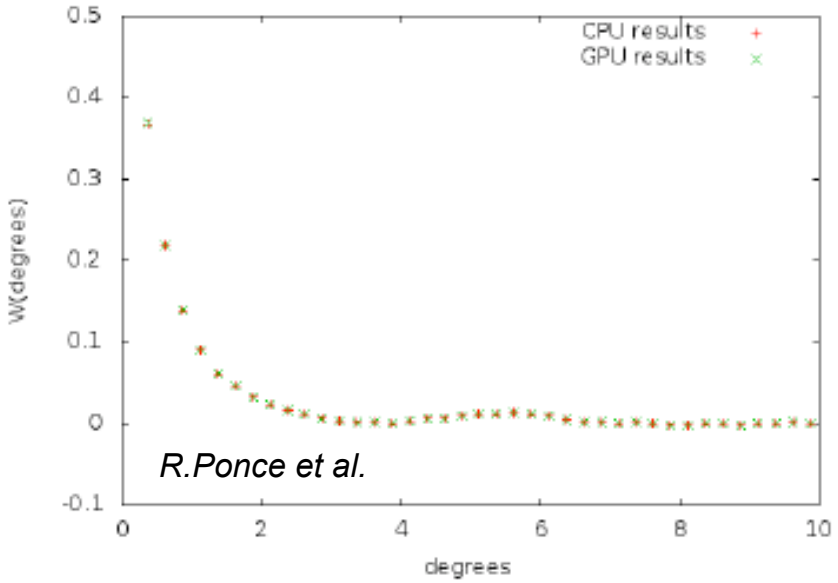
M.Cárdenas et al.

Intel Xeon 4-cores against 3 GPU models

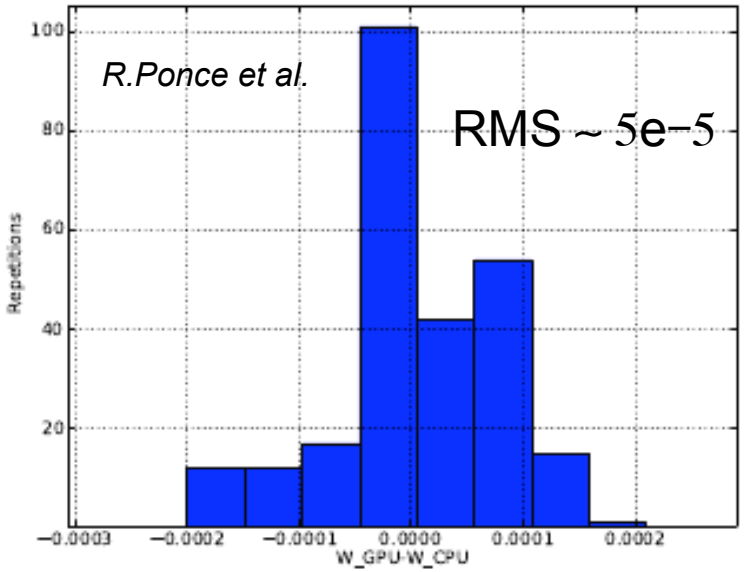
Ciemat

Results: residuals negligible wrt statistical errors

Superposed w(θ)



W_GPU(θ) – W_CPU(θ)



Residuals are smaller using latest GPU cards (single → double precision)

Ciemat

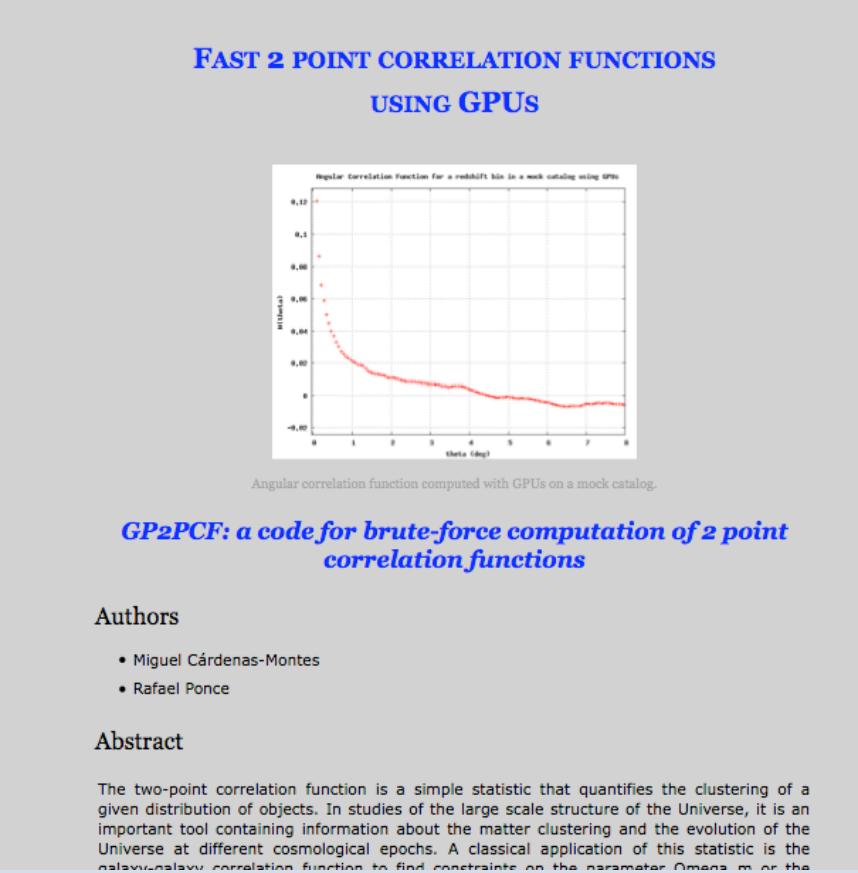
There are some constraints when developing for GPUs

- Steep learning curve (not so if used to parallel processing)
- Applications limited by memory available at GPU
- Binning constraints

Presenting: GP2pcf

We provide source code in CUDA, documentation and support.

Tarball with code an examples available at:



FAST 2 POINT CORRELATION FUNCTIONS USING GPUS

Angular correlation function computed with GPUs on a mock catalog.

GP2PCF: a code for brute-force computation of 2 point correlation functions

Authors

- Miguel Cárdenas-Montes
- Rafael Ponce

Abstract

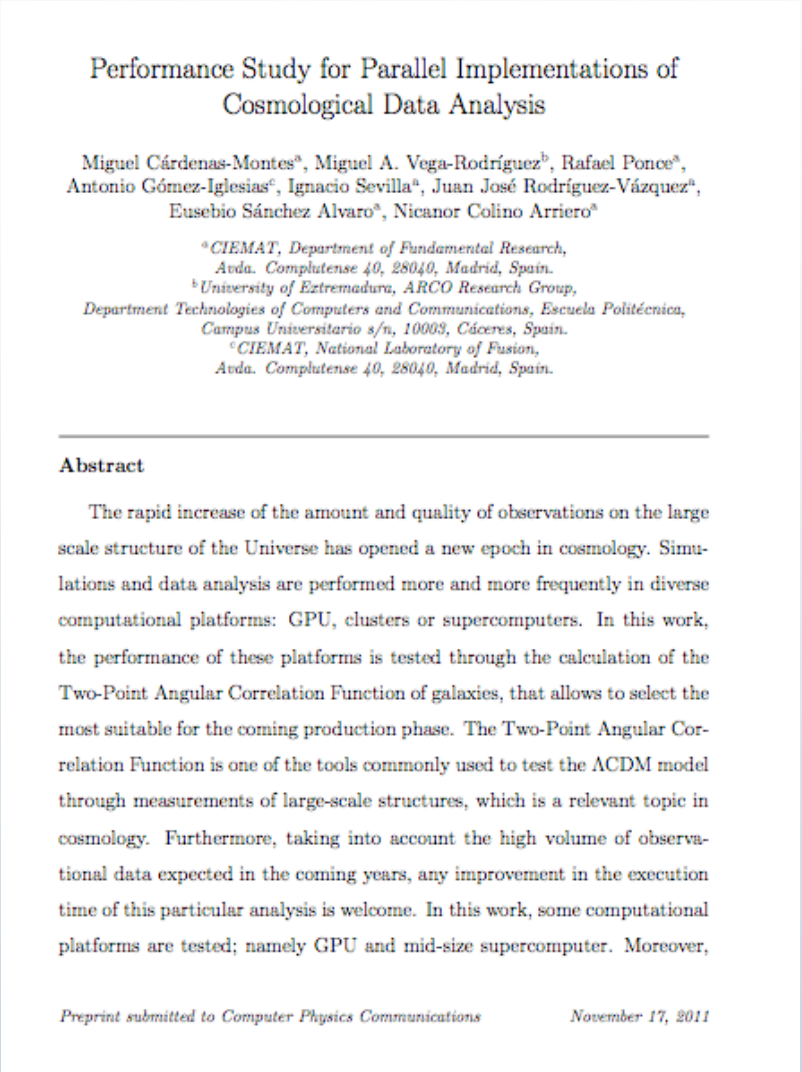
The two-point correlation function is a simple statistic that quantifies the clustering of a given distribution of objects. In studies of the large scale structure of the Universe, it is an important tool containing information about the matter clustering and the evolution of the Universe at different cosmological epochs. A classical application of this statistic is the

http://wwwae.ciemat.es/cosmo/gp2pcf

Notes on the implementation available

A preprint describing the code and tests has been submitted to Computer Physics Communications. **Available on request!**

Also check out XXI ADASS proceedings (directly available for download from GP2PCF website).



Performance Study for Parallel Implementations of Cosmological Data Analysis

Miguel Cárdenas-Montes[a], Miguel A. Vega-Rodríguez[b], Rafael Ponce[a], Antonio Gómez-Iglesias[c], Ignacio Sevilla[a], Juan José Rodríguez-Vázquez[a], Eusebio Sánchez Alvaro[a], Nicanor Colino Arriero[a]

[a]*CIEMAT, Department of Fundamental Research, Avda. Complutense 40, 28040, Madrid, Spain.*
[b]*University of Extremadura, ARCO Research Group, Department Technologies of Computers and Communications, Escuela Politécnica, Campus Universitario s/n, 10003, Cáceres, Spain.*
[c]*CIEMAT, National Laboratory of Fusion, Avda. Complutense 40, 28040, Madrid, Spain.*

Abstract

The rapid increase of the amount and quality of observations on the large scale structure of the Universe has opened a new epoch in cosmology. Simulations and data analysis are performed more and more frequently in diverse computational platforms: GPU, clusters or supercomputers. In this work, the performance of these platforms is tested through the calculation of the Two-Point Angular Correlation Function of galaxies, that allows to select the most suitable for the coming production phase. The Two-Point Angular Correlation Function is one of the tools commonly used to test the ΛCDM model through measurements of large-scale structures, which is a relevant topic in cosmology. Furthermore, taking into account the high volume of observational data expected in the coming years, any improvement in the execution time of this particular analysis is welcome. In this work, some computational platforms are tested; namely GPU and mid-size supercomputer. Moreover,

Preprint submitted to Computer Physics Communications *November 17, 2011*

Ciemat

Future developments

- k-tree instead of brute-force approach
- galaxy-shear correlation
- N-point correlation function
- **3D correlation function**
- **multi-GPU**
- **pyCUDA version**

Ciemat

Other uses in cosmology

- Matching catalogs
- K-nearest neighbors (e.g. photo-z) (V. García)
- N-body simulations (C. Frigaard)
- ...

Summary

GPUs are **inexpensive** and **powerful** tools for **2-point correlation function** computation. The cost is lower in equivalent chips, Watts/Gflop, personnel, fight for resources.

The **code** to profit from this is **available**:

http://wwwae.ciemat.es/cosmo/gp2pcf

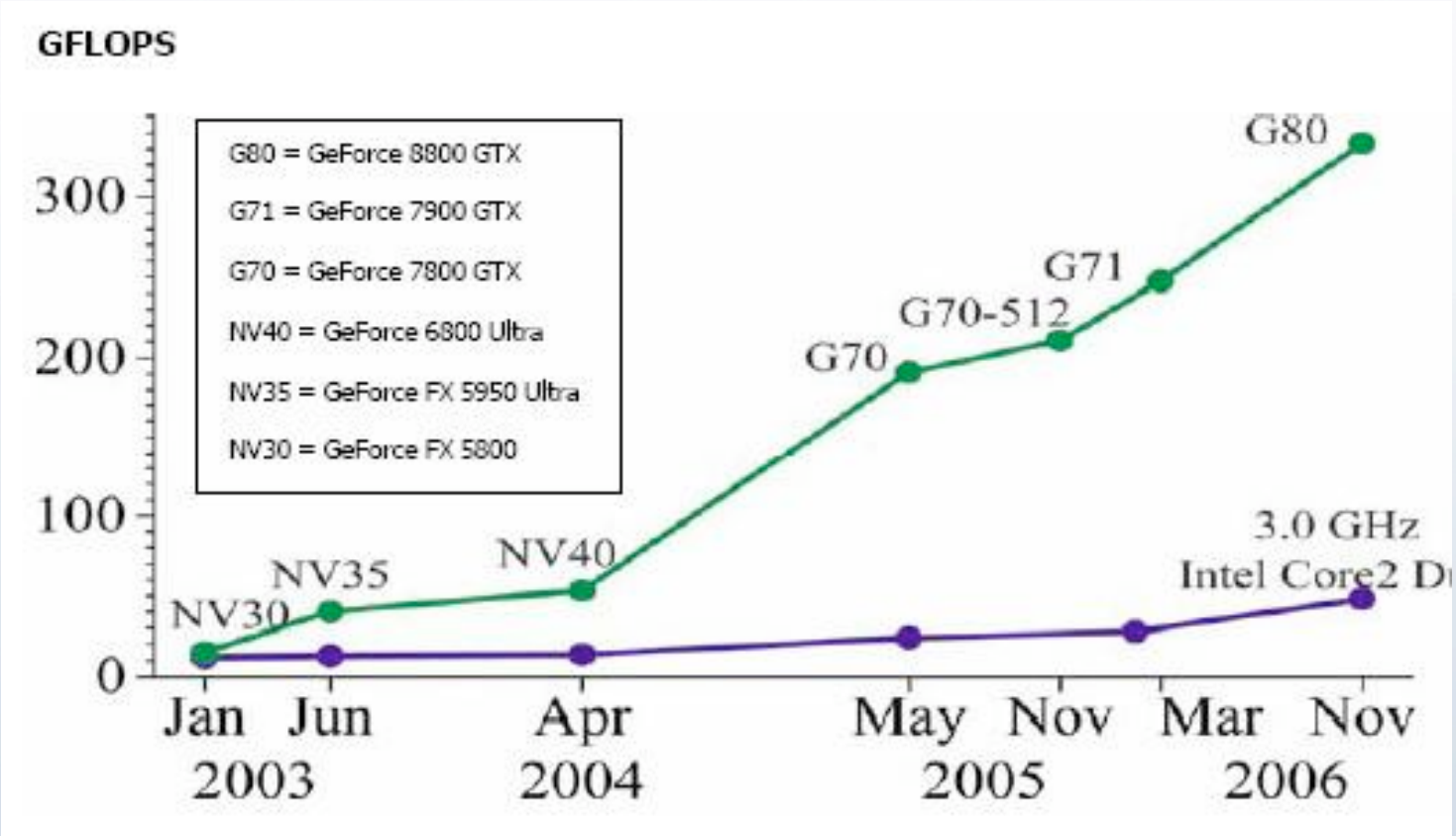
Summary

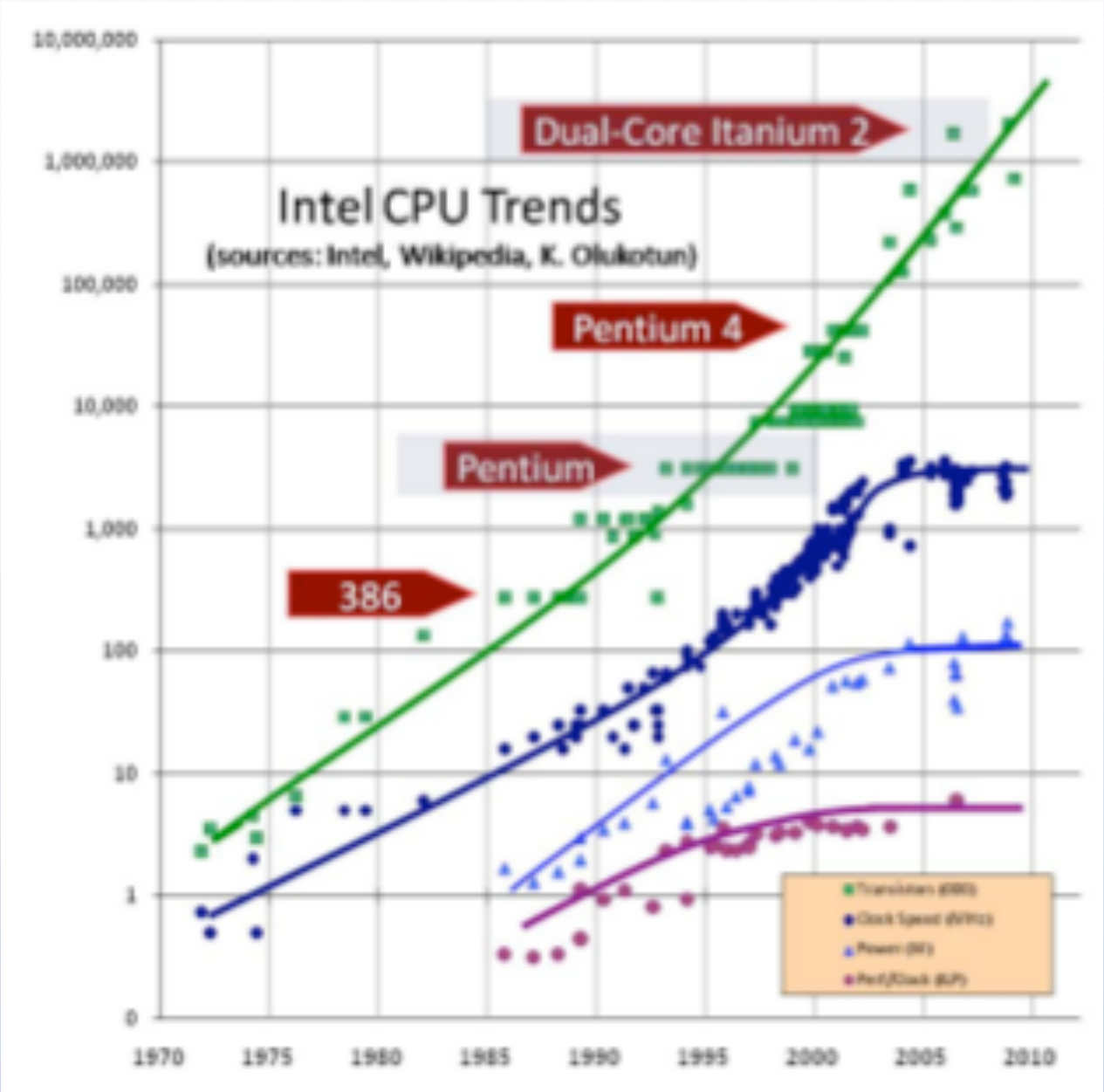
GPUs are **inexpensive** and **powerful** tools for **2-point correlation function** computation.

The **code** to profit from this is **available**:

http://wwwae.ciemat.es/cosmo/gp2pcf

"Supercomputing for the masses" *Dr. Dobbs*


GFLOPS
G80 = GeForce 8800 GTX
G71 = GeForce 7900 GTX
G70 = GeForce 7800 GTX
NV40 = GeForce 6800 Ultra
NV35 = GeForce FX 5950 Ultra
NV30 = GeForce FX 5800
300
200
100
0
G80
G71
G70-512
G70
NV40
NV35
NV30
3.0 GHz
Intel Core2 D
Jan
Jun
Apr
May
Nov
Mar
Nov
2003
2004
2005
2006


Intel CPU Trends
(sources: Intel, Wikipedia, K. Olukotun)
Dual-Core Itanium 2
Pentium 4
Pentium
386
10,000,000
1,000,000
100,000
10,000
1,000
100
10
1
0
1970
1975
1980
1985
1990
1995
2000
2005
2010