

# Casos de Éxito en Computación con GPUs

Miguel Cárdenas Montes

CIEMAT

Madrid, 1 Octubre 2014

# Table of Contents

Introduction

2PACF

Hardware

Shear

Conclusions

# Introduction

## Motivation for this Presentation

To show some success cases in GPU computing, including drawbacks and benefits.

# Introduction

## Motivation for this Presentation

To show some success cases in GPU computing, including drawbacks and benefits.

## Motivation for GPU Computing

To analyse or simulate the high volume of data in Cosmology in a reasonable processing time.

# 2PACF

## Cosmology-Astronomy Facing Data Challenge



# Introduction

## Why GPU Computing?

- GPU has a good relation FLOPS per watt.
- It is not expensive, at least not more than the budget for computing in cosmology.
- It can cover the computational need of a mid-sized group.
- Two entries (2nd Titan and 6th Piz Daint) in the top 10 of Top500, and 10 entries in Green500.
- It is a hot topic (publications).

# Introduction

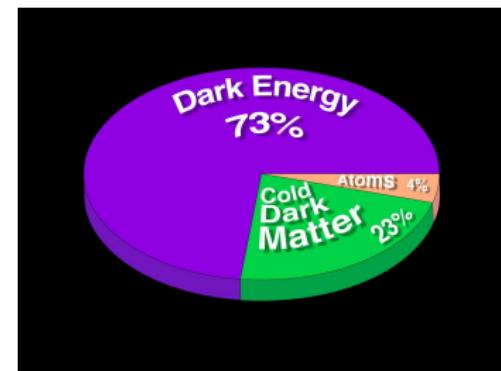
This is not:

- A presentation about the NVIDIA hardware (I am not NVIDIA staff),
- nor a CUDA training,
- but our own experience when trying to solving scientific problems with GPU-CUDA.

# First Project: The Two-Point Angular Correlation Function

# 2PACF

- The distribution of galaxies in the universe is one of the most important probes for cosmological models.
- The 2PACF is a test to measure this distribution.
- The calculation of the 2PACF is computationally demanding,  $O(N^2)$ . CPU implementation takes around 8 hours for a sample of 430K galaxies.



## Data Volume

Statistical Analysis from  $10^6$  to  $10^{10}$  galaxies in the near future.

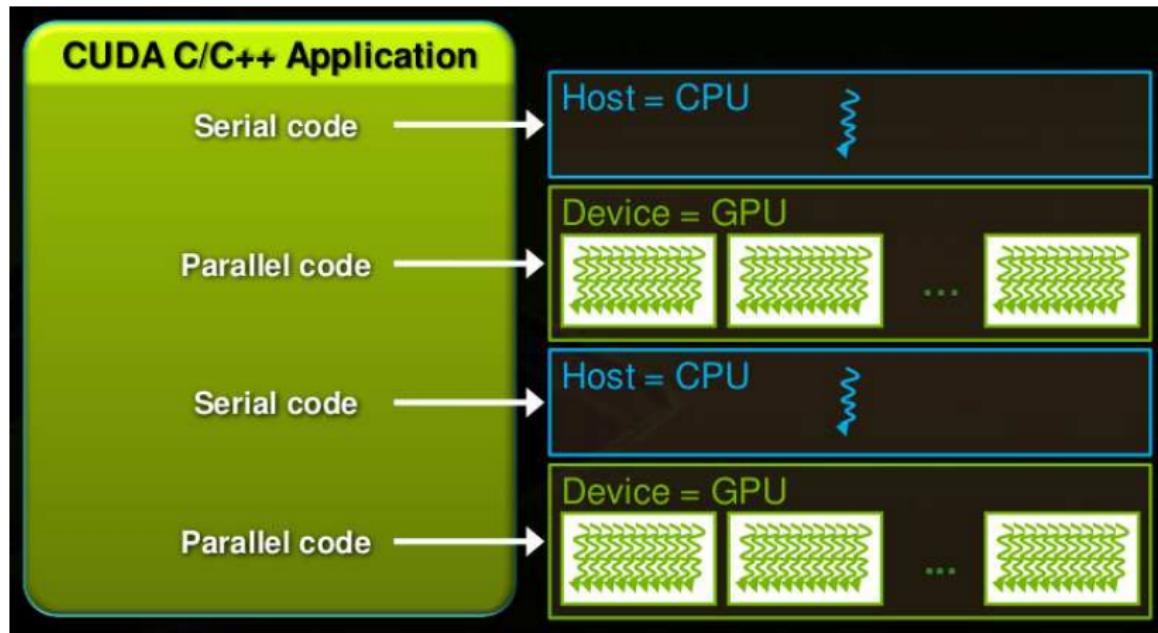
# 2PACF

Our goals:

- Fast Implementation of Two-Point Angular Correlation Function. CPU implementation takes around 8 hours for a sample of 430K galaxies, and  $O(N^2)$ !
- An implementation able to deal with very large surveys,  $> 10^6$  galaxies.
- Low budget.



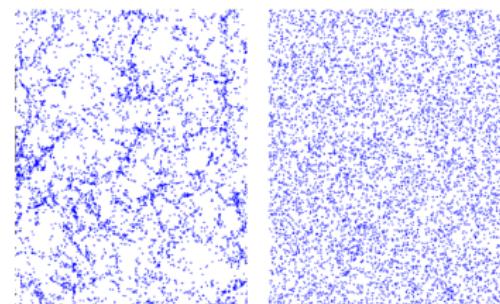
# Code Structure



## 2PACF

The 2PACF,  $\omega(\theta)$ , is a measure of the excess or lack of probability of finding a pair of galaxies under a certain angle with respect to a random distribution. In general, estimators 2PACF are built by combining the following quantities:

- $DD(\theta)$  is the number of pairs of galaxies for a given angle  $\theta$  chosen from the data catalogue (D).
- $RR(\theta)$  equivalent on the random catalogue (R).
- $DR(\theta)$  one galaxies from each catalogue.



# 2PACF

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

- If  $\omega(\theta) > 0$  more frequently found at angular separation of  $\theta$  than expected for a randomly distributed.
- If  $\omega(\theta) < 0$  lack of galaxies in this particular  $\theta$ .
- $\omega(\theta) = 0$  means purely random distribution.
- The sample 430K galaxies.

# GPU Implementation

- Intense use of *shared memory* for intermediate calculations and histogram construction.
- Sub-histograms allocated on *shared memory*.
- Atomic operations on shared memory also required.
- Coalesced-access to global memory. Data ordered by coordinates, not by galaxies.

# Execution Time

**Table:** Execution time and speedup for several implementations.

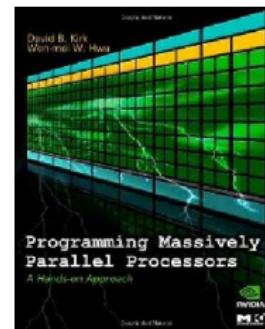
Implementation	Execution time (s)	Speedup
CPU	35,186.327	
OpenMP (8 cores+hyperthreading)	3,326.363	10
GPU (GTX295)	305.570	115
MPI 64 cores	403.937	87
MPI 128 cores	205.008	171

Cárdenas-Montes, Miguel, et al.: **New Computational Developments in Cosmology**, Ibergrid, 101-112, 2012

Ponce, Rafael, et al.: **Application of GPUs for the Calculation of Two Point Correlation Functions in Cosmology**, Astronomical Data Analysis Software and Systems XXI, 461, 73-76, 2012

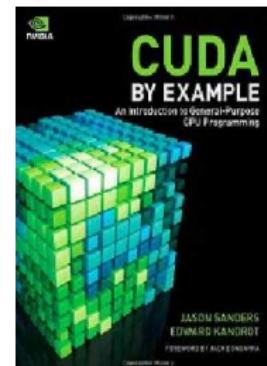
# Drawbacks

- No expert in programming.
- No didactic books. No training events.  
Lack of information in web.
- No atomicAdd() for float, only integer.  
This is essential for some other analyses  
in Cosmology, specially in correlation  
functions.
- Single precision.



## More Optimization - New Card

- New card, C2075.
  - AtomicAdd() in float.
  - Double precision.
- New books (very didactic).



## More Optimization - New Card

- Initial implementation:
  - Coalesced-access to data in global memory.
  - Intense use of *shared memory* for intermediate calculations and histogram construction.
  - Atomic operations on shared memory also required: main bottleneck.
- Improvements:
  - Register for data frequently reused, incrementing the data locality.
  - Incrementation of the occupancy by reducing the number of variable and, therefore, forcing to recalculate.
  - When possible reducing branching by replacing if-conditionals by min-functions.

# GPU Implementation

Implementation			
Single Precision, Compute Capability 1.2, GTX295	Original Code	$299,566.4 \pm 15.3$ ms	
	Positive Strategies	$275,303.8 \pm 17.6$ ms	
	<b>Reduction</b>	<b>24,262.6</b>	
	<b>Speedup</b>		<b>1.09</b>
Single Precision, Compute Capability 2.0, C2075	Original Code	$314,346.8 \pm 199.6$ ms	
	Positive Strategies	$269,969.5 \pm 69.8$ ms	
	<b>Reduction</b>	<b>44,377.3</b>	
	<b>Speedup</b>		<b>1.16</b>
Double Precision, Compute Capability 2.0, C2075	Original Code	$452,097.3 \pm 287.2$ ms	
	Positive Strategies	$438,934.0 \pm 132.2$ ms	
	<b>Reduction</b>	<b>13,163.4</b>	
	<b>Speedup</b>		<b>1.03</b>

# Outcome

- Satisfactory level of performance to deal with larger files  $> 10^6$  galaxies even more.
- Other problem proposed.

Cárdenas-Montes, Miguel, et al.: **Calculation of Two-Point Angular Correlation Function: Implementations on Many-Core and Multicore Processors**, Ibergrid, Editorial Universitat Politecnica de Valencia, 203-214, 2013

# A Bit of Hardware

# TESLA C2075 Fermi 2.0

- 14 streaming multiprocessors (SM);  
32 cuda cores / SM = total of 448 cuda cores;  
32 threads / core = total of 14336 threads
- clock rate: 1.15GHz
- 6 GB memory (global memory)
- 64KB on-chip memory  
SM and 768KB L2 cache (shared by all SMs)



XEON E7-8870

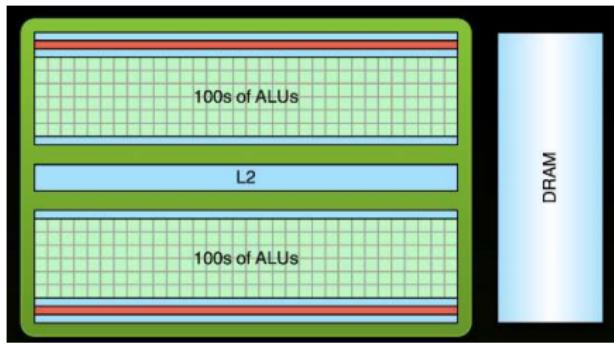
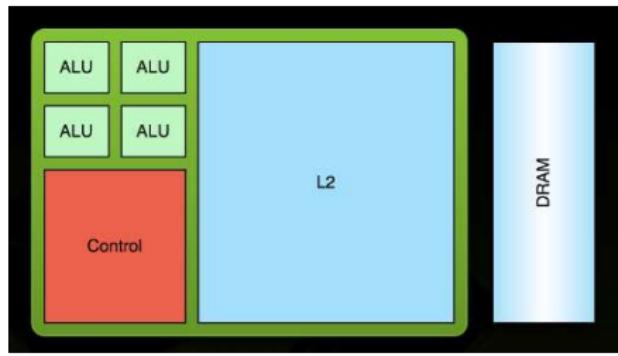


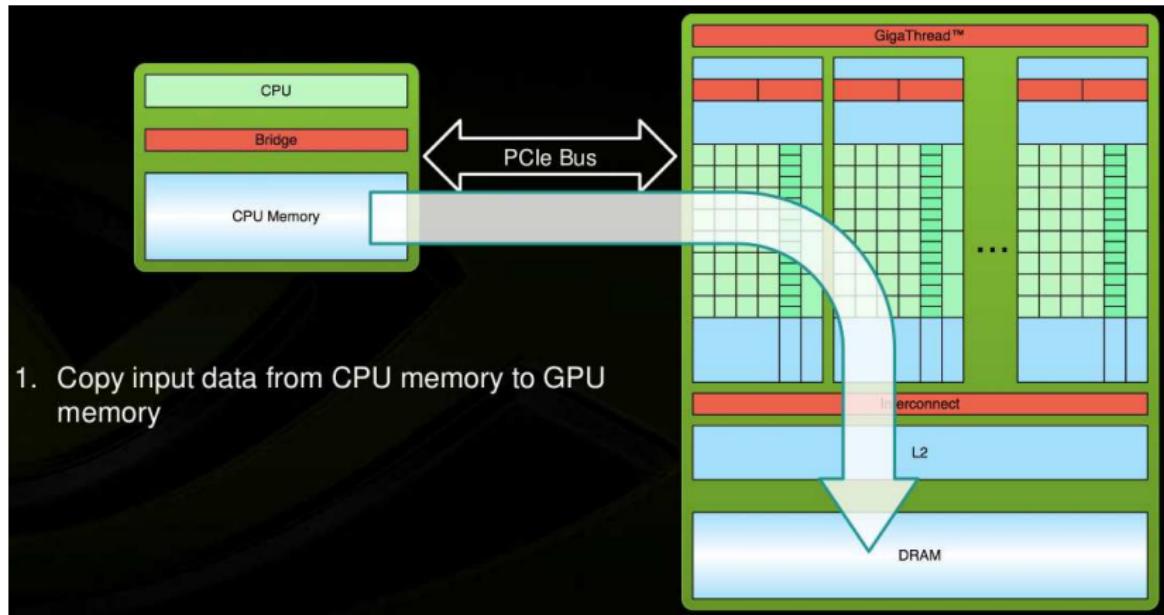
- 10 cores / 20 threads
- 2.40GHz
- 30MB cache

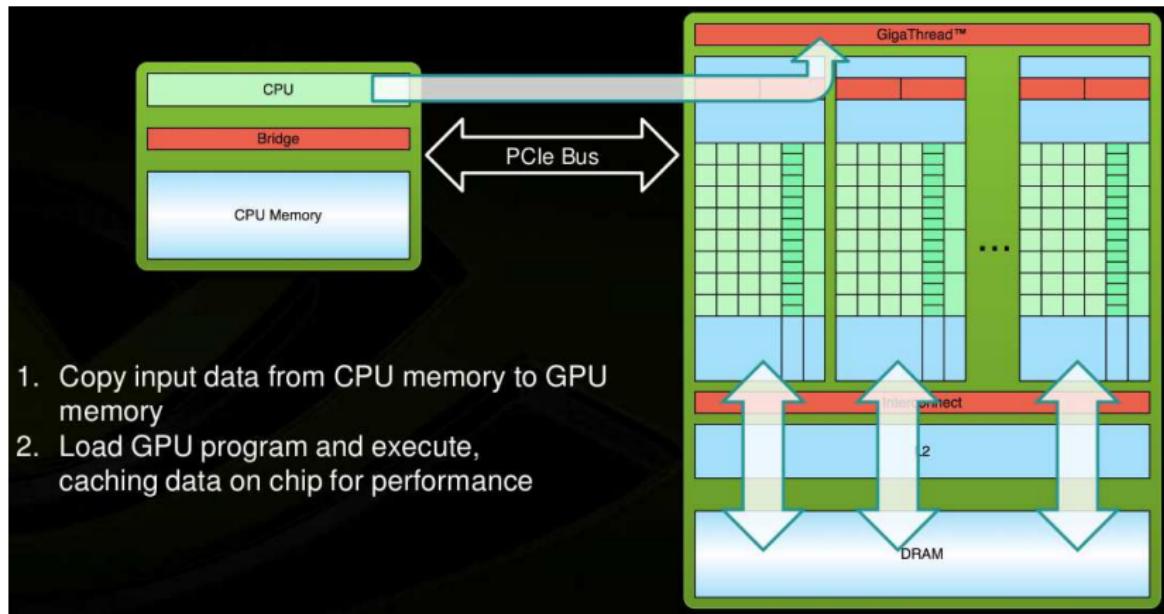
i7-3970X

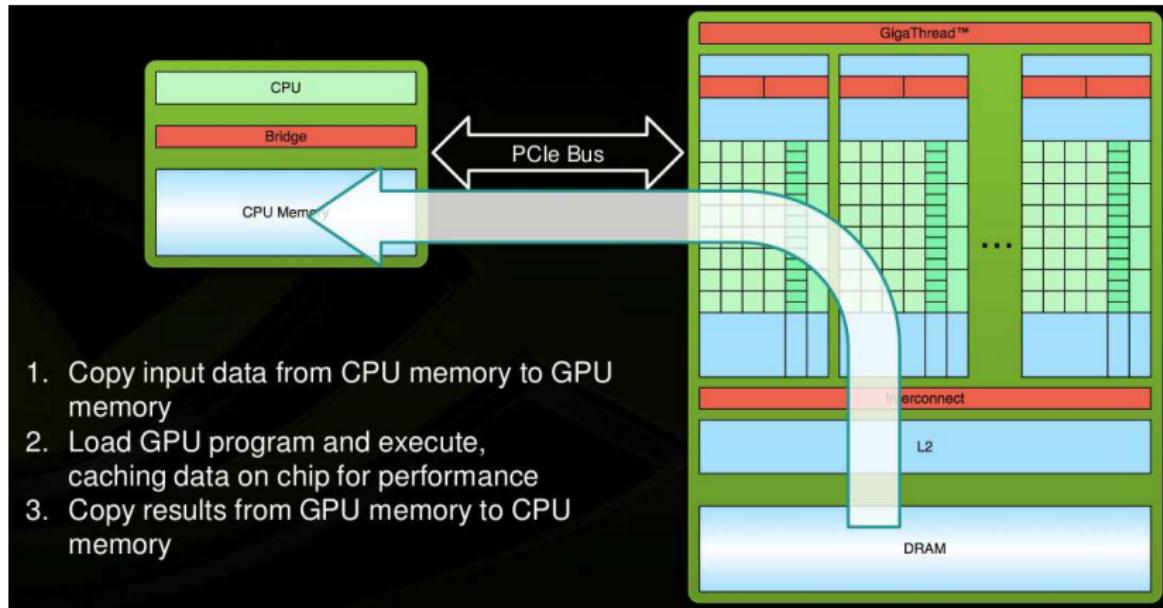


- 6 cores / 12 threads
- 3.50GHz
- 15MB cache

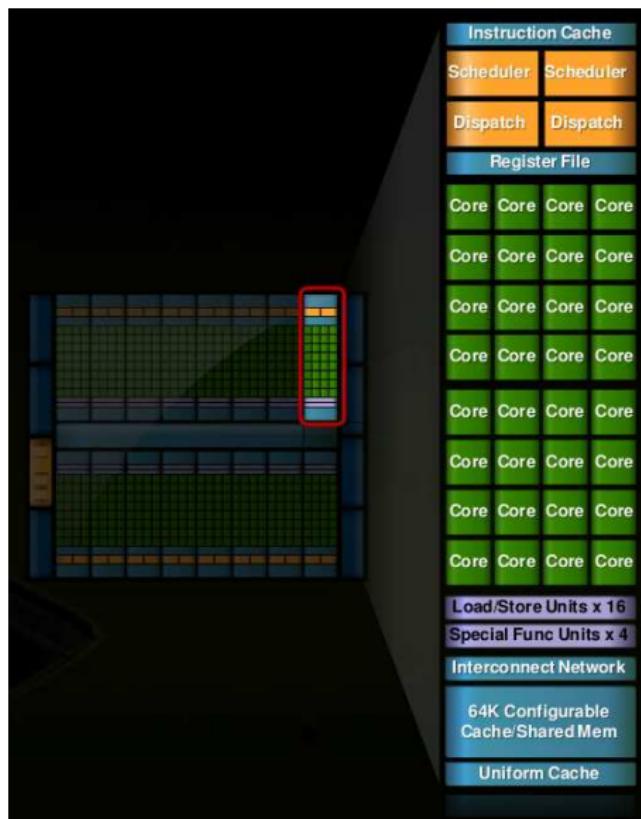


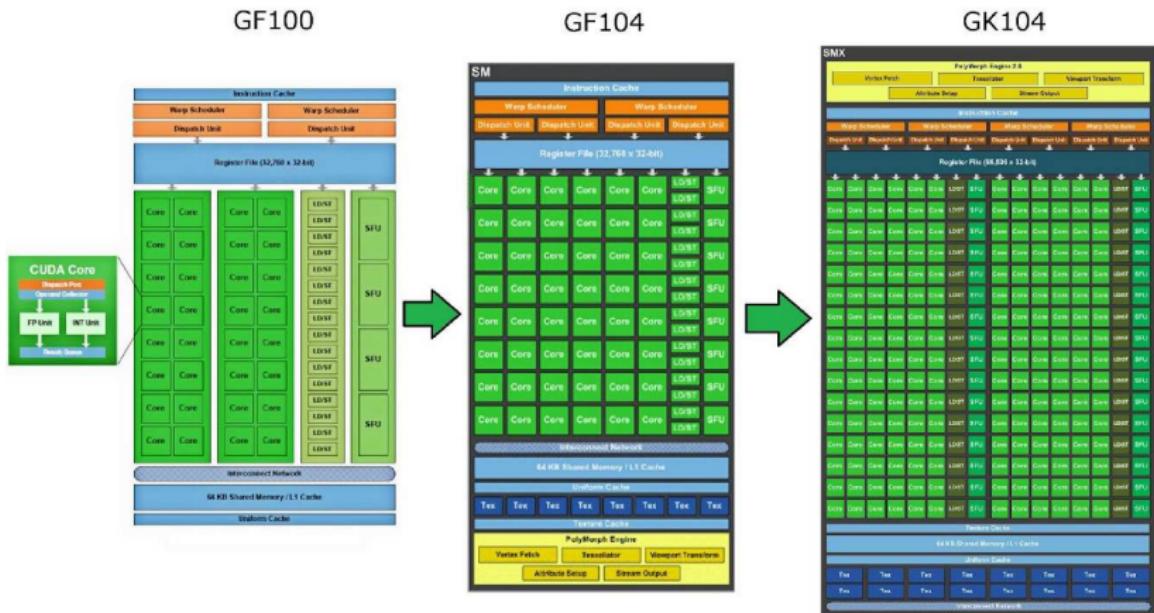












# Global Memory Bandwidth

- Many-core processors have limited off-chip memory access bandwidth compared to peak compute throughput
- Fermi
  - 1 TFLOPS SPFP peak throughput
  - 0.5 TFLOPS DPFP peak throughput
  - 144 GB/s peak off-chip memory access bandwidth
    - 36 G SPFP operands per second
    - 18 G DPFP operands per second
  - To achieve peak throughput, a program must perform  $1,000/36 = \sim 28$  arithmetic operations for each operand value fetched from off-chip memory

# Global Memory (DRAM) Bandwidth

Ideal



Reality



©Wen-mei W. Hwu and David Kirk/NVIDIA  
Barcelona, July 2-6, 2012

3

# Shear-Shear Correlation Function

# Shear-Shear Correlation

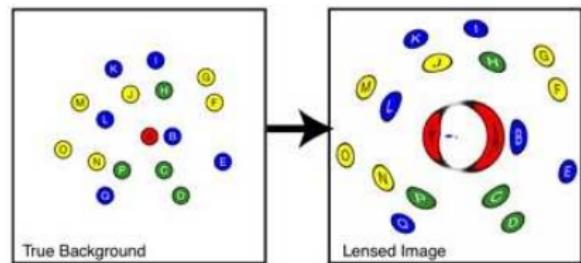
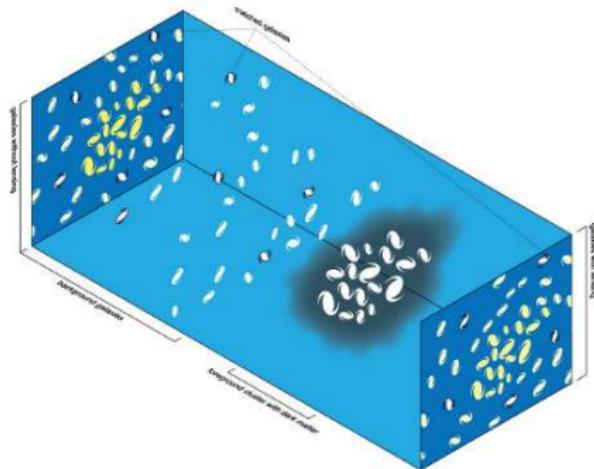
The physics of the problem:

- Light rays are deflected when travelling through a gravitational potential, this phenomenon is known as gravitational lensing.
- This causes the observed shapes of distant galaxies to be very slightly distorted by the intervening matter in the Universe, as their light travels towards us. This distortion is called *cosmic shear*.
- By measuring this component it is possible to derive the properties of the mass distribution causing the distortion.

# Shear-Shear Correlation

- In the past this analysis has been burden by instrumental errors, reduced volume of data and the available observational data span small regions of the sky.
- Data volume from tens of thousands to tens of millions.
- In this work an observational data set of 1 million of galaxies (Canada–France–Hawaii Lensing Survey, CFHTLenS).
- Shear-shear has a higher computational intensity than 2PACF (more calculation for each pair of galaxies).

# Shear-Shear Correlation



# Shear-Shear Correlation

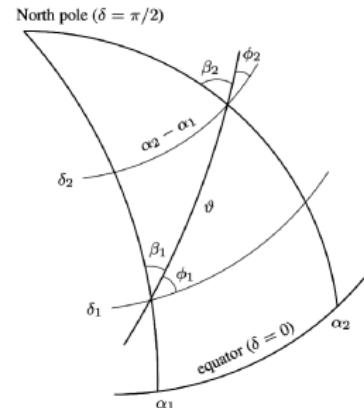
$$\cos \Phi_1 = \frac{\sin(\alpha_2 - \alpha_1) \cos \delta_2}{\sin \theta}$$

$$\sin \Phi_1 = \frac{\cos \delta_2 \sin \delta_1 - \sin \delta_2 \cos \delta_1 \cos(\alpha_2 - \alpha_1)}{\sin \theta}$$

$$\xi_+(\theta) = \frac{\sum_{ij} w_i w_j (\gamma_t(\theta_i) \cdot \gamma_t(\theta_j) + \gamma_x(\theta_i) \cdot \gamma_x(\theta_j))}{\sum_{ij} w_i w_j}$$

$$\xi_-(\theta) = \frac{\sum_{ij} w_i w_j (\gamma_t(\theta_i) \cdot \gamma_t(\theta_j) - \gamma_x(\theta_i) \cdot \gamma_x(\theta_j))}{\sum_{ij} w_i w_j}$$

$$\xi_x(\theta) = \frac{\sum_{ij} w_i w_j (\gamma_t(\theta_i) \cdot \gamma_x(\theta_j))}{\sum_{ij} w_i w_j}$$



# New Weapons



# Base Implementation

The best practices are inherited:

- A coalesced pattern access to global memory.
- An intensive use of shared memory to store the results of intermediate operations is implemented.
- The use of registers to store the input data frequently accessed (such as galaxy coordinates and ellipticities).
- Sub-histogram construction on shared memory and final gathering on global memory.

## Base Implementation

And other best practices are learned now:

- Use of double precision. Very tiny quantities are added in the histograms (in the 2PACF an unit added); and the galaxies are analysed at very small angles.
- AtomicAdd() in float required.
- Explicitly caching global memory into shared memory, L1 cache memory off.

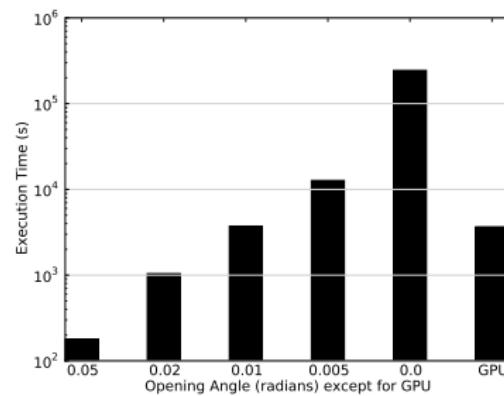
# Compute capability

- Compute capability 1.1: Atomic function only in global memory.
- Compute capability 1.2: Atomic function in shared memory.
- Compute capability 1.3: Some functions from single to double precision (IEEE 754-1985).

# Shear-Shear Correlation

Comparison with the previous state-of-the-art code, ATHENA.

- ATHENA is a sequential code based on kd-trees to reduce the computational complexity of the calculations.
- Based on the parameter termed *Opening Angle, OA*. The smaller OA it is, the fewer approximations makes.



## Outcome

Execution time: GPU 3,618.7s vs. ATHENA 247,681s at OA=0.

Cárdenas-Montes, Miguel, et al.: **GPU-Based Shear-Shear Correlation Calculation**, Computer Physics Communications, 185(1):11-18, ISSN: 0010-4655, 2014

# Shear-Shear Correlation

Extra optimization applied:

Implementation	Execution Time (s)	Speedup Related to the Baseline	Speedup Related to ATHENA OA=0
Baseline	3,618.7		67
Reordered loops	3,243.3	1.12	75
Vectorized	3,184.2	1.14	77

# MPI-CUDA Implementation

MPI-CUDA Implementation to deal with tens of million of galaxies (1M in this table):

Nodes	Execution Time	Speedup Related to MPI-CUDA Single-Node	Speedup Related to ATHENA OA=0
1	3,325.39		73.4
2	1,672.59	1.99	145.9
4	845.15	3.93	288.7
8	432.24	7.69	564.6
16	225.49	14.75	1082.2

# MPI-CUDA Implementation

## Outcome

- 15M galaxies in the GPU implementation, the execution time takes 169 hours,
- MPI-CUDA implementation with 16 nodes it takes 11 hours, achieving a speedup of 15.36.

Cárdenas-Montes, Miguel, et al.: **High-Performance Implementations for Shear-Shear**

**Correlation Calculation.** Cluster, IEEE Computer Society, 2014. Aceptado como póster.

# Conclusions

# Conclusions

- GPU-CUDA a mature technology.
- Soundness of the science provided. More and more disciplines on it.
- Jump forward other capabilities: analyses and simulations.

# Thanks

Gracias