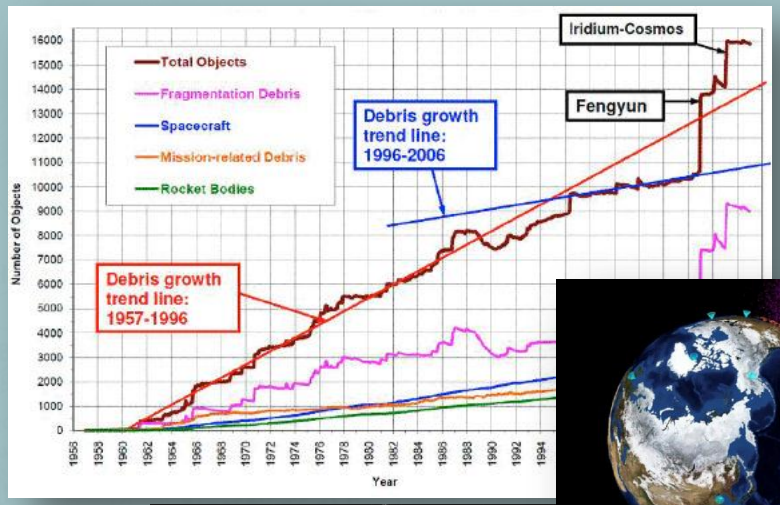
OPEN CL ACCELERATED SIMPLIFIED GENERAL PERTURBATIONS 4 ALGORITHM

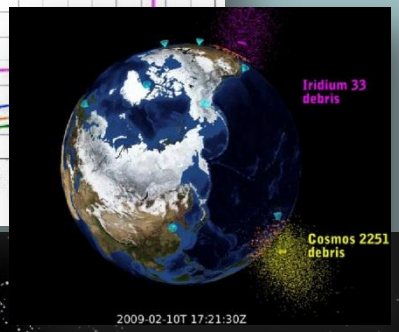
Juan Andres Fraire, Pablo Ferreyra, and Carlos Marques

PROBLEM DESCRIPTION

- The number of space objects such as satellites, spacecraft, and debris are increasing significantly.
  - 2007: Chinese anti-satellite missile test
  - 2009: Iridium-33 collision with Cosmos
- ISS crew forced to run to escape capsule due to prediction failure

Need for better tracking and prediction systems!





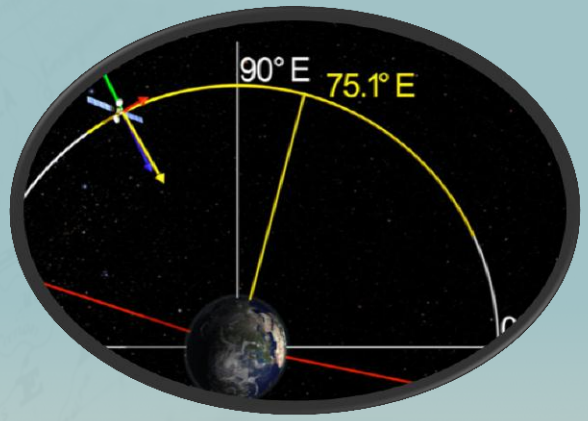
CURRENT APPROACH

More Tracking Stations

Too Expensive!

Space Tracking organizations:
Air Force Space Command (AFSCN)
&
Others…
Permanently track and predicts space activity

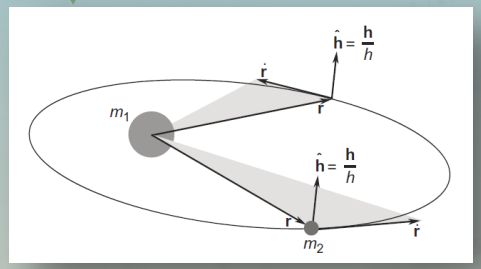
Better Propagation Mechanisms

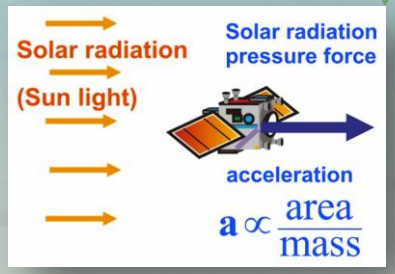


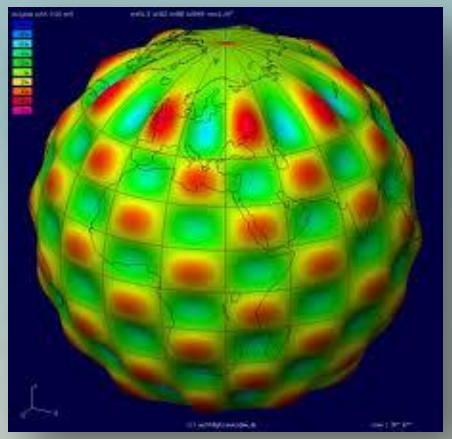
We go for this one!

SATELLITE PROPAGATORS

- Basically, they tell us when and where an orbiting object will be in the future
- Several perturbations affects objects trajectories:
  - Non-uniform gravitation field ("J" Coefficients)
  - Atmospheric Drag
  - Sun Radiation Pressure
  - 3rd Body (Sun, Moon, Venus, etc)







SATELLITE PROPAGATORS

- Different propagators accounts for different perturbations effects:

- **Two Body**
- **J2 Propagator**
- **SGP4**
- **HPOP**

- Earth Gravitation Force
- Secular gravitation effects
- 3rd Body (Moon and Sun)
- Resonance Effects
- Simple Atmospheric Drag Model
- Near Earth & Deep Space Model

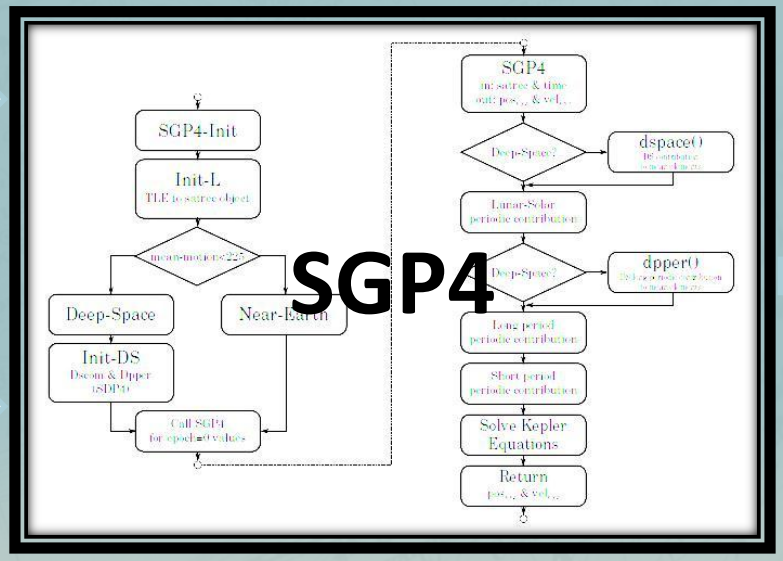
"Publicly available and widely used because of it good performance/precision for tracking purposes"

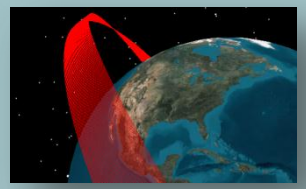
SGP-4 PROPAGATOR

- SGP4 Algorithm implies complex trigonometric calculations, but essentially:

Initial Orbital Parameters (**TLE**)

Time from Initial parameters (**T**)





Position Vector
&
Velocity Vector
In time **T**

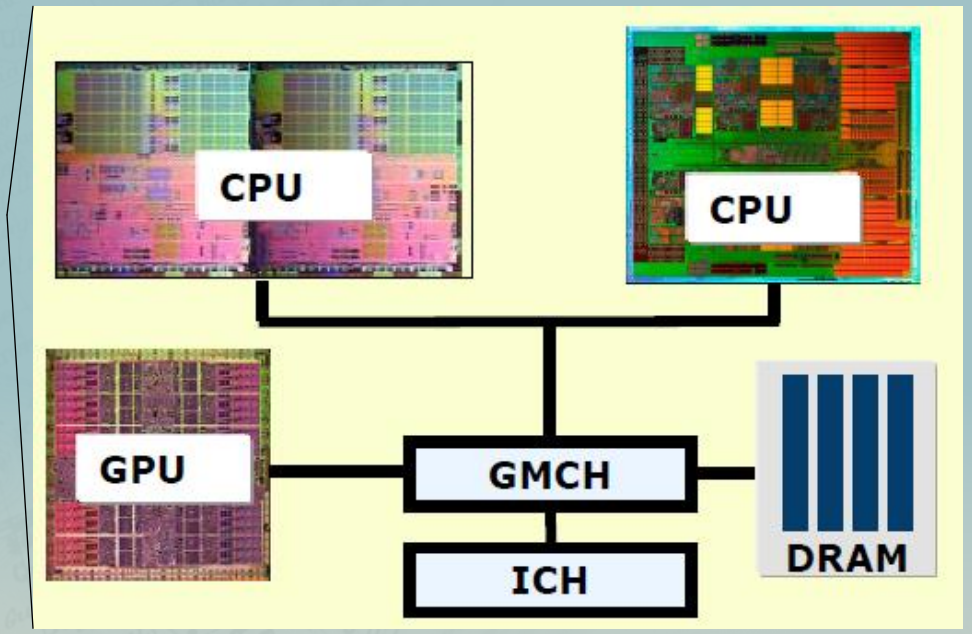
“Earth Centered Inertial Coordinate Reference System”

SGP-4 PROPAGATOR IN OPEN-CL

- We implemented this algorithm in ***Open-CL*** to improve the execution performance in an ***heterogeneous system***

"A collection of CPUs, GPUs, DSP, FPGA, and any processing element in a given platform"

Several orbiting objects propagated In parallel in many devices!



GMCH = graphics memory control hub,
ICH = Input/output control hub

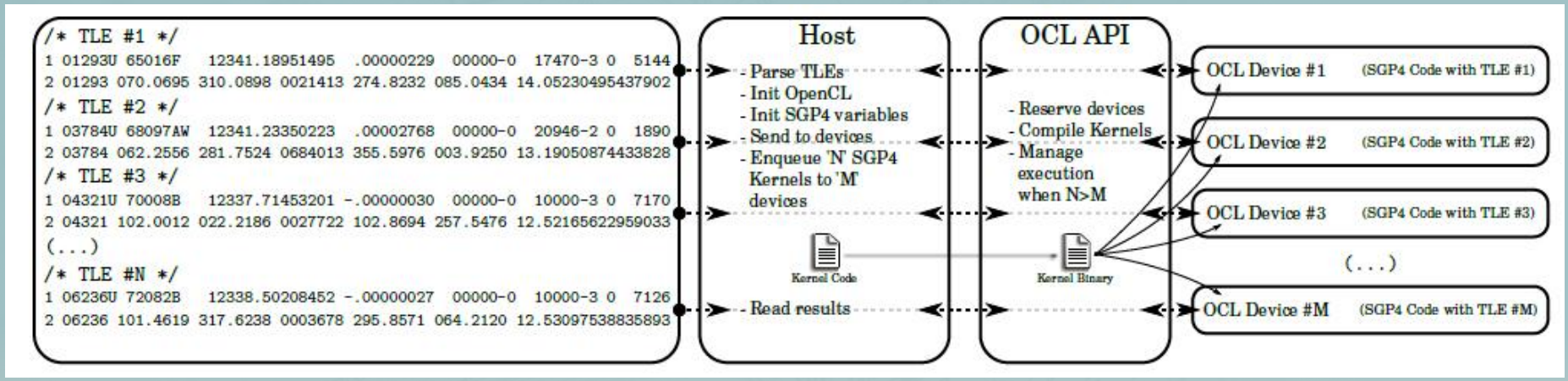
ABOUT OPEN-CL

- Open-CL is a **cross-platform**, **cross-operating system**, **cross-vendor open standard**
- Oriented to programming a collection of **CPUs, GPUs, DSPs, FPGAs**, and others, even **cell-phones** and **mainframes**!
- Allows to assign "**tasks**" (named Kernels) to the most proper device
- Exploits **System** and **Device** parallelism at the same time

HOW WE DID IT

TLE Files defines the initial conditions of many orbiting objects

An Open-CL Host Initializes and distribute the work load to available hardware. It also read the results and store or show in the screen



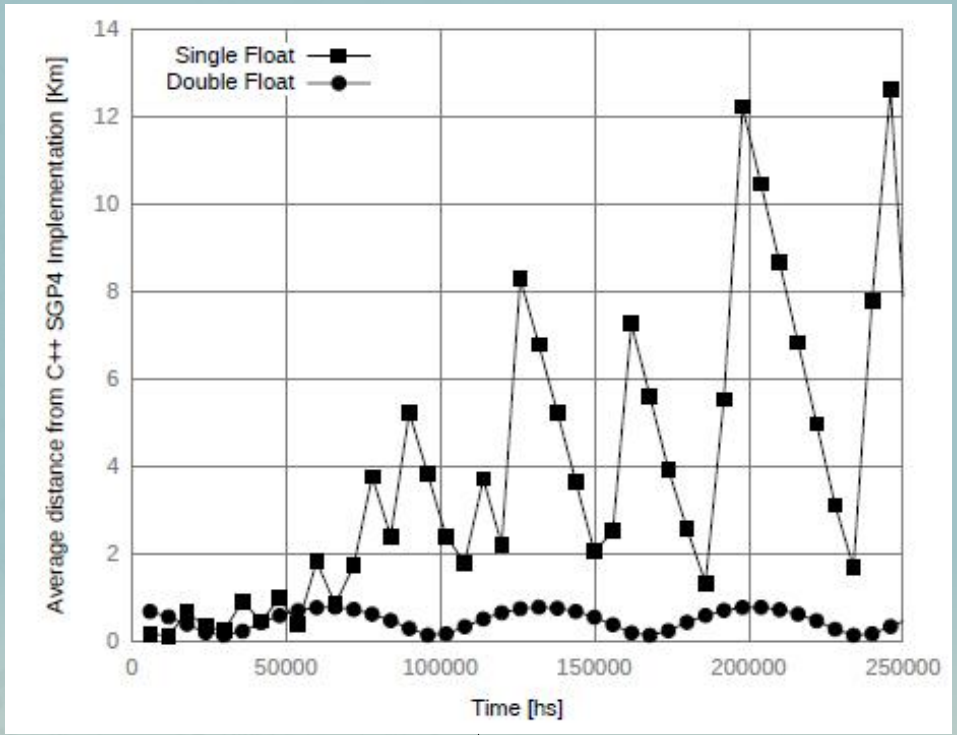
OpenCL APIs are available for Linux, Windows, and OSX

We tested on CPUs and GPUs devices

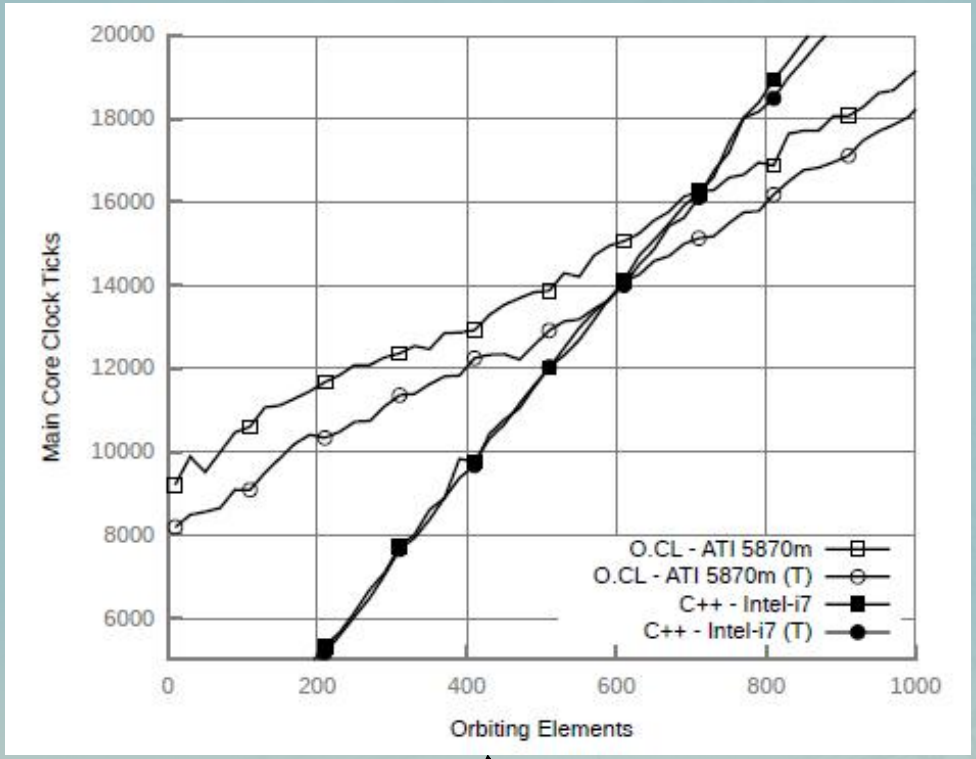
WHAT WE CONSIDERED: PRECISION

- Depending on the available device, double floating point calculation might or not be available…

We provide two versions of SGP-4 Kernel to better adapt the underlying hardware with no considerable measurements error



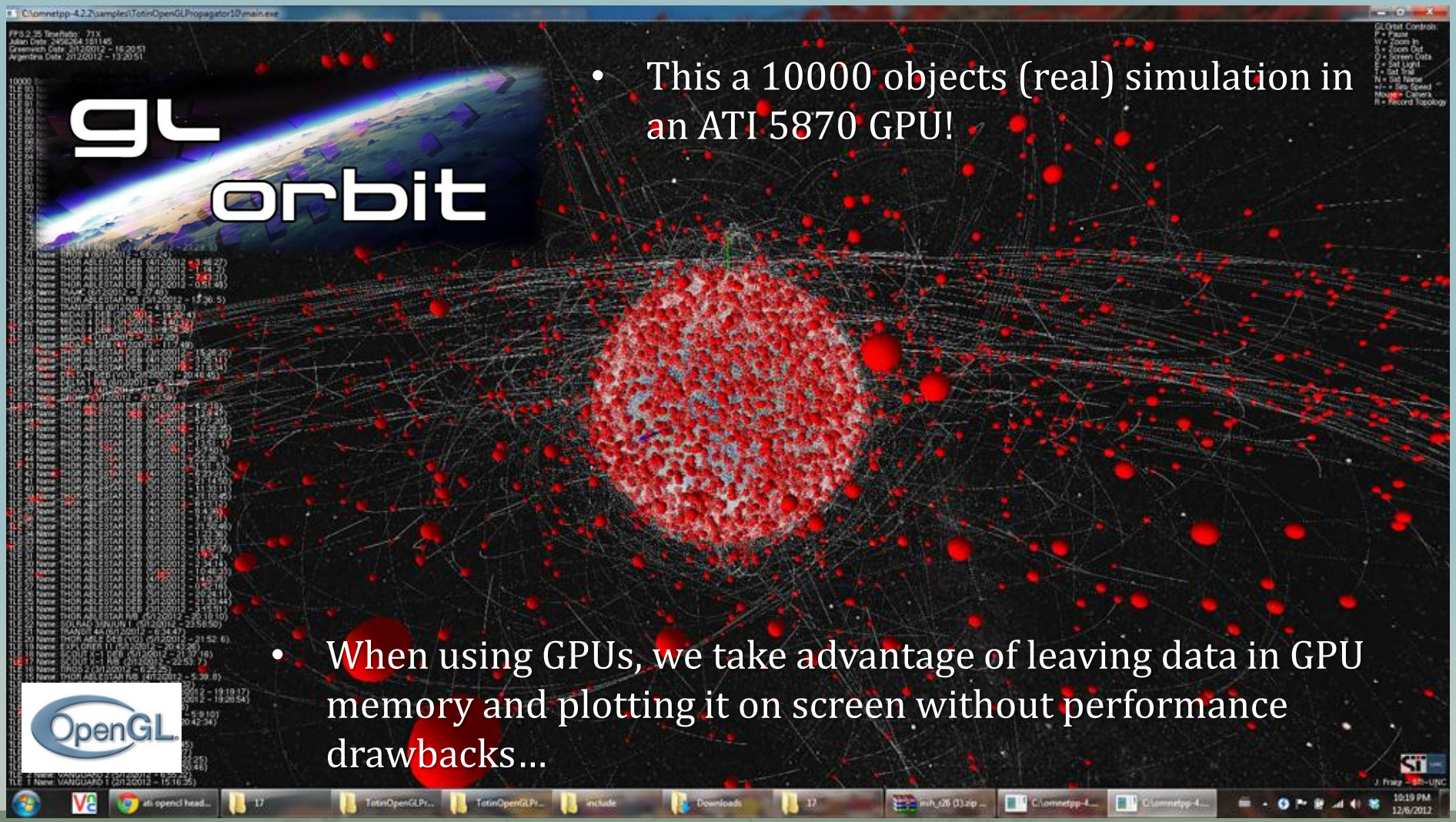
WHAT WE CONSIDERED: TIME



- Also we allow to advance time (**T**) within the same Kernel or from the Host Code

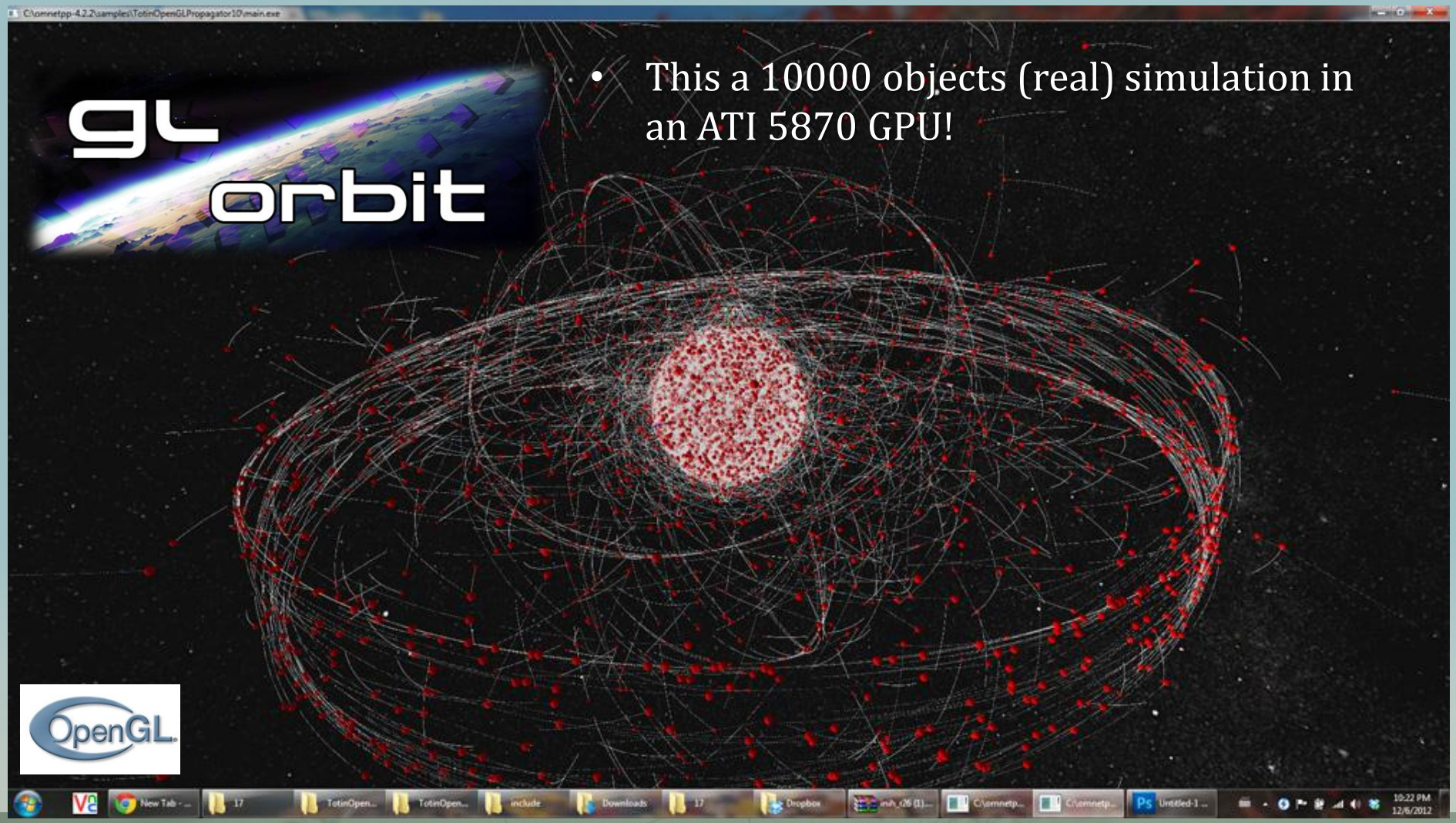
Advancing the time in the Kernel limits propagation control but delivers increasing performance as it drastically reduces memory transfers to the device

WHAT WE CONSIDERED: GRAPHICS



- This a 10000 objects (real) simulation in an ATI 5870 GPU!
- When using GPUs, we take advantage of leaving data in GPU memory and plotting it on screen without performance drawbacks…

WHAT WE CONSIDERED: GRAPHICS



- This a 10000 objects (real) simulation in an ATI 5870 GPU!

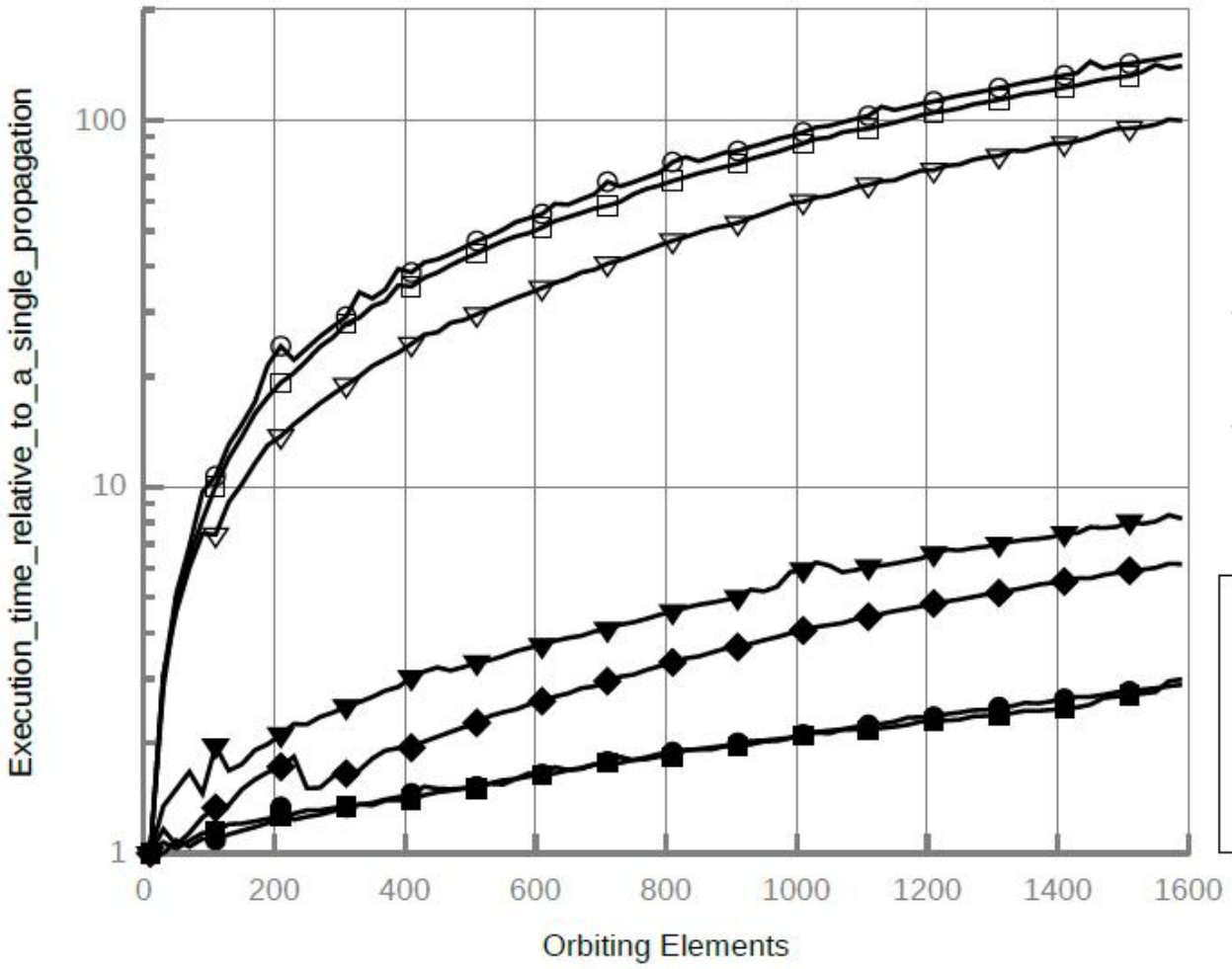
PERFORMANCE EVALUATION

- As Open-CL executes in almost **any underlying hardware**, we evaluated our implementation in several processing devices:

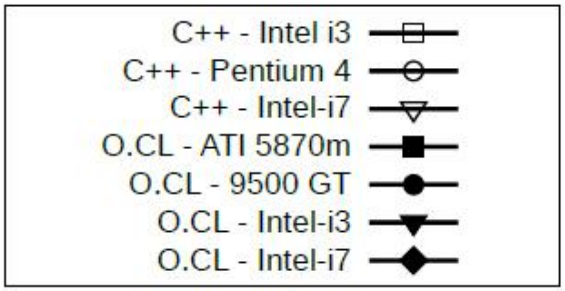
Vendor	Device	Type	Platform Version	Platform Vendor	Comp. Units	Max Dim.	Max WI in a Dim.	Max WG Size	Max Clk Freq	Addr. Bits	Global Mem Size	Local Mem Size	Double F. Point
Intel(R)	Core(TM) i7 CPU Q 720 @ 1.60GHz	CPU	OpenCL 1.1 AMD-APP-SDK-v2.5	Advanced Micro Devices, Inc.	8	3	1024	1024	1596MHz	32	2GB	32KB	Yes
Intel(R)	Core(TM) i3-2330M CPU @ 2.20GHz	CPU	OpenCL 1.1 LINUX	Intel(R) Corporation	4	3	1024	1024	2200MHz	64	4GB	32KB	Yes
AMD(R)	ATI HD 5870 Mobility	GPU	OpenCL 1.1 AMD-APP-SDK-v2.5	Advanced Micro Devices, Inc.	10	3	256	256	700MHz	32	1GB	32KB	No
NVIDIA(R)	GeForce 9500 GT	GPU	OpenCL 1.1 CUDA 4.1.1	NVIDIA Corporation	4	3	512	512	1350MHZ	32	512MB	16KB	No

- We propose a **1600** orbiting element satellite scenario with a position call each hour for each element for a complete year ⇨ this is **8760*1600** kernels executions!

RESULTS



Open-CL outperforms standard non-parallel optimized C++ compilations in all cases, while extending the available hardware to perform the propagation tasks



CONCLUSION

- We implemented the **SGP-4** Algorithm in **Open-CL**
- This allows to better use powerful COTS hardware in a **transparent** yet **efficient** fashion
- No significant calculation errors are evidenced for considering double floating point precision

- **Performance improvement** is always welcomed in an ever **increasing debris tracking system**

GL-ORBIT VIDEO