Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Cosmica: GPU accelerated CR propagation code

Giovanni Cavallotto (INFN MiB), Stefano Della Torre (INFN MiB)

Spoke 3 General Meeting, Elba 5-9 / 05, 2024

Scientific Rationale

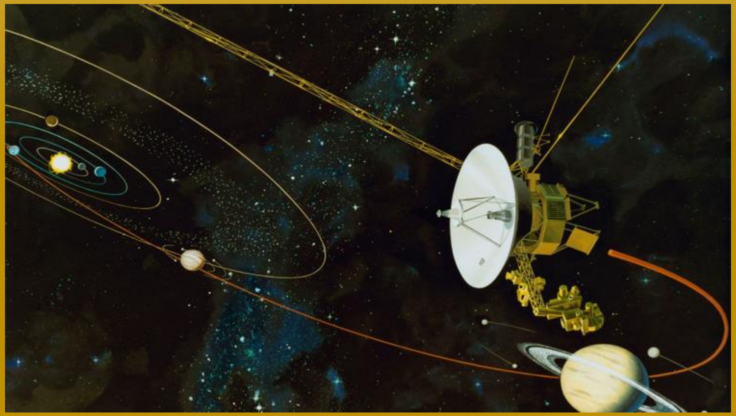
Galactic Cosmic Rays modulation inside the heliosphere

(region dominated by solar wind and related phenomena)

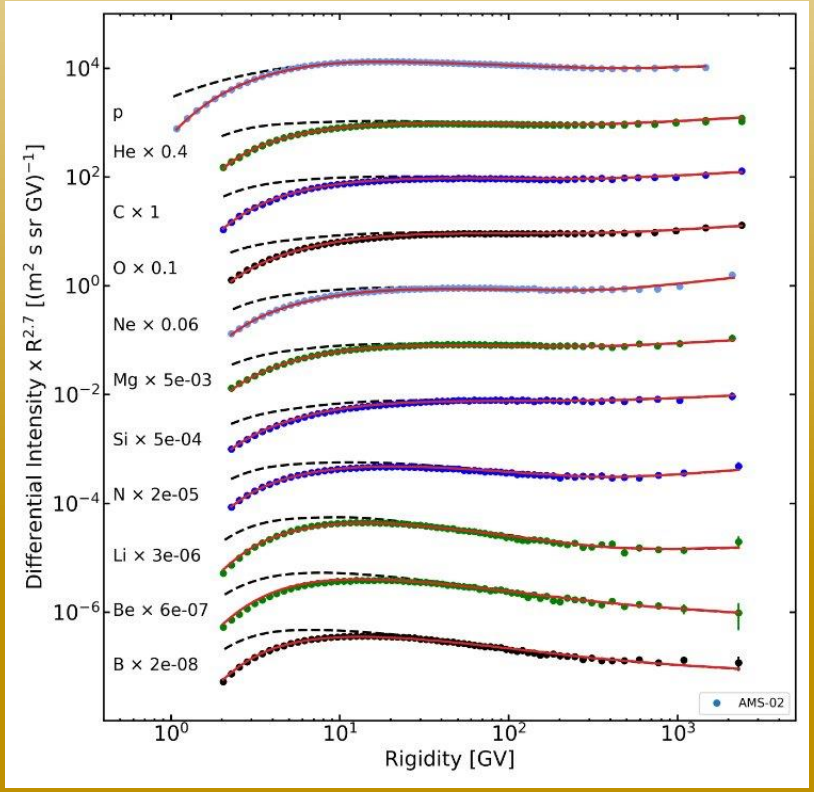
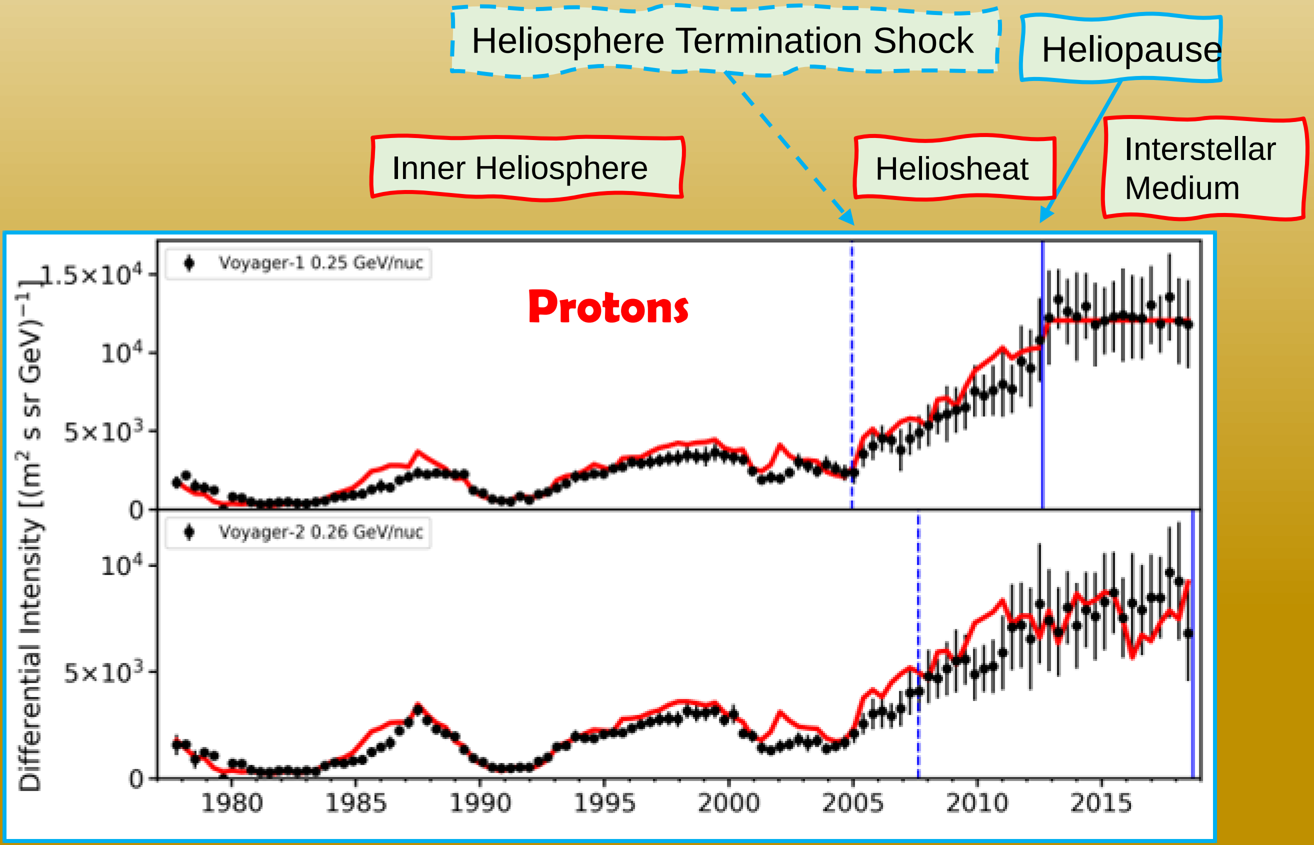
- **CR astrophysics (heliospere, propagation & sources modelling)**
- **Space weather (space radiation environment)**
- **Single event effect (device demage)**
- **CR background in space experiments**

AMS-02 (Earth orbit)

Voyager (Heliosphere boundaries)

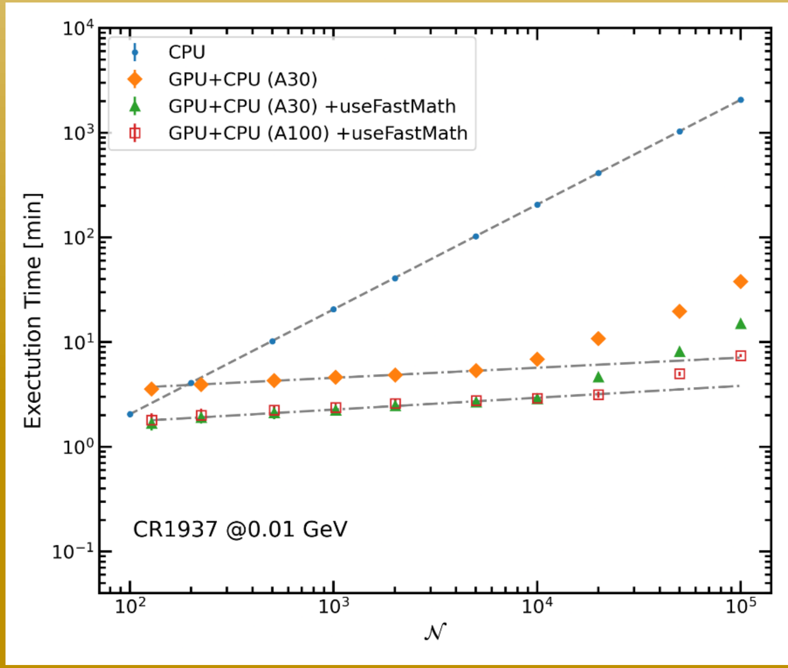


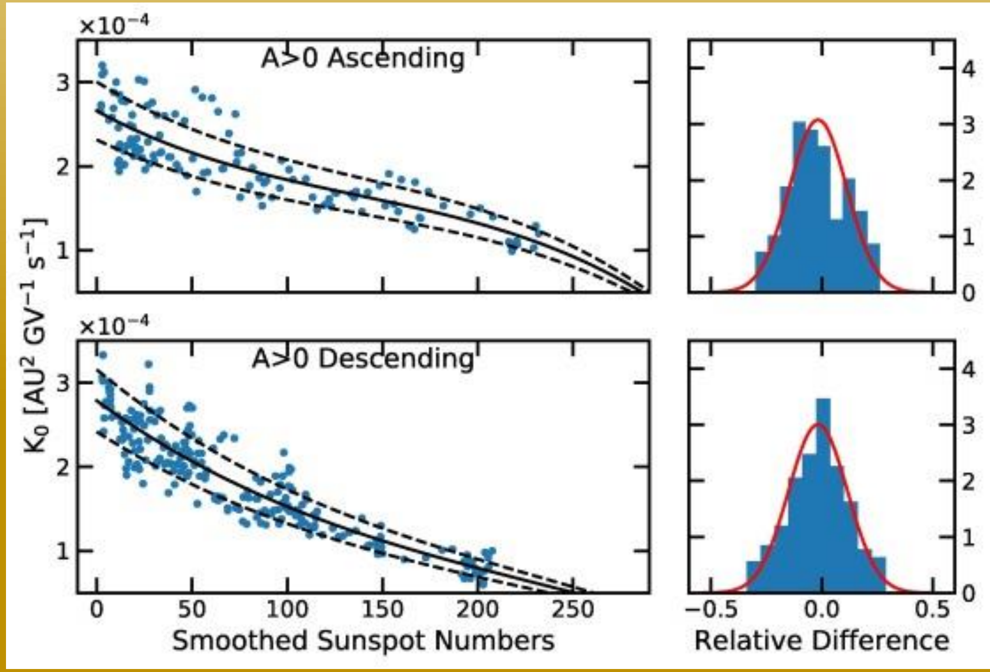
Scientific Rationale

Local CR modulation	Time evolution along an orbit
	

Technical Objectives

- Complete GPU porting of the SDE integration code (Cosmica)
- Code optimization based on the NVIDIA Ampere architecture (from hours to few minutes execution time)
- Implement different physical models as libraries and test physical parameters





Methodologies

Diffusion — Advective — Adiabatic loss — Drift

$$\frac{\partial U}{\partial t} = \frac{\partial}{\partial x_i}\left(K^S_{ij}\frac{\partial U}{\partial x_j}\right) + \frac{1}{3}\frac{\partial V_{sw,i}}{\partial x_i}\frac{\partial}{\partial T}\left(\alpha_{rel} T U\right) - \frac{\partial}{\partial x_i}\left[\left(V_{sw,i} + v_{d,i}\right)U\right]$$

Parker Transport Equation

Fokker-Plank (Kolmarov) differential equation

↓ Ito's formula

$$\frac{\partial U}{\partial s} = \sum_i A_{B,i}(s,y)\frac{\partial U}{\partial y_i} + \frac{1}{2}\sum_{i,j} C_{B,ij}(s,y)\frac{\partial^2 U}{\partial y_i\,\partial y_j} - L_B U + S$$

Stochastic Differential Equations

(MC backward in time numerically integrated)

↓ Particle propagation indipendence

SIMD parallelization on GPU

CPU-GPU mixed CUDA code

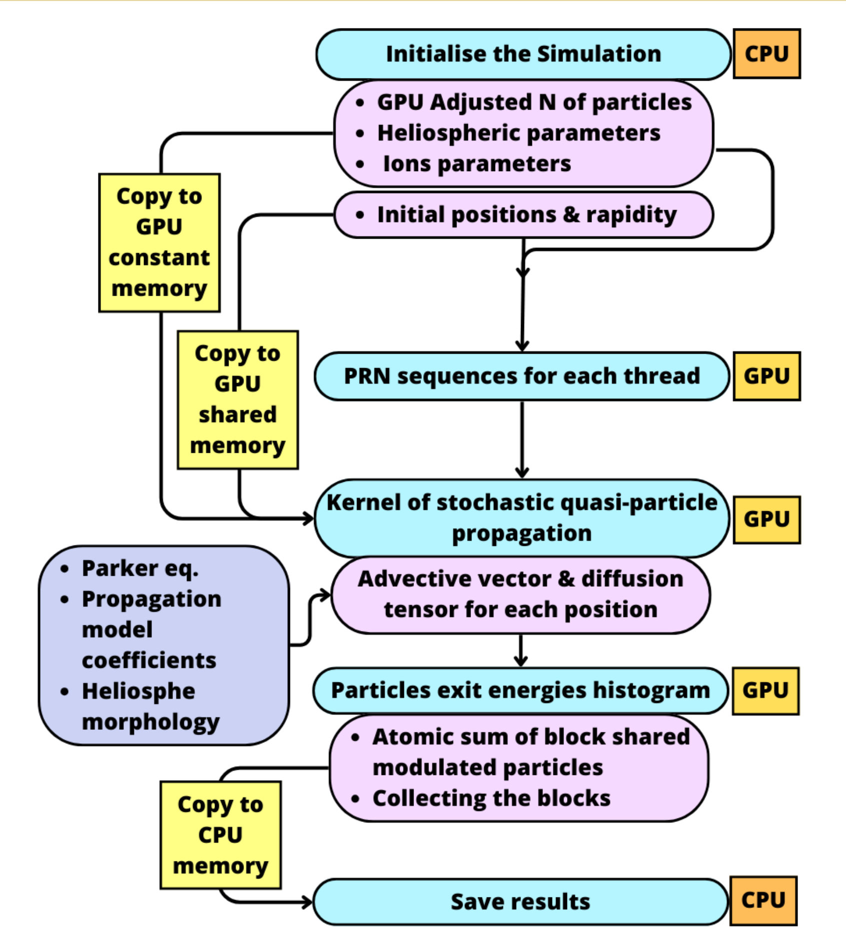
→

Everage of N quasi-particles simulated for each input energy

Convolved with Local Interstellar Spectrum

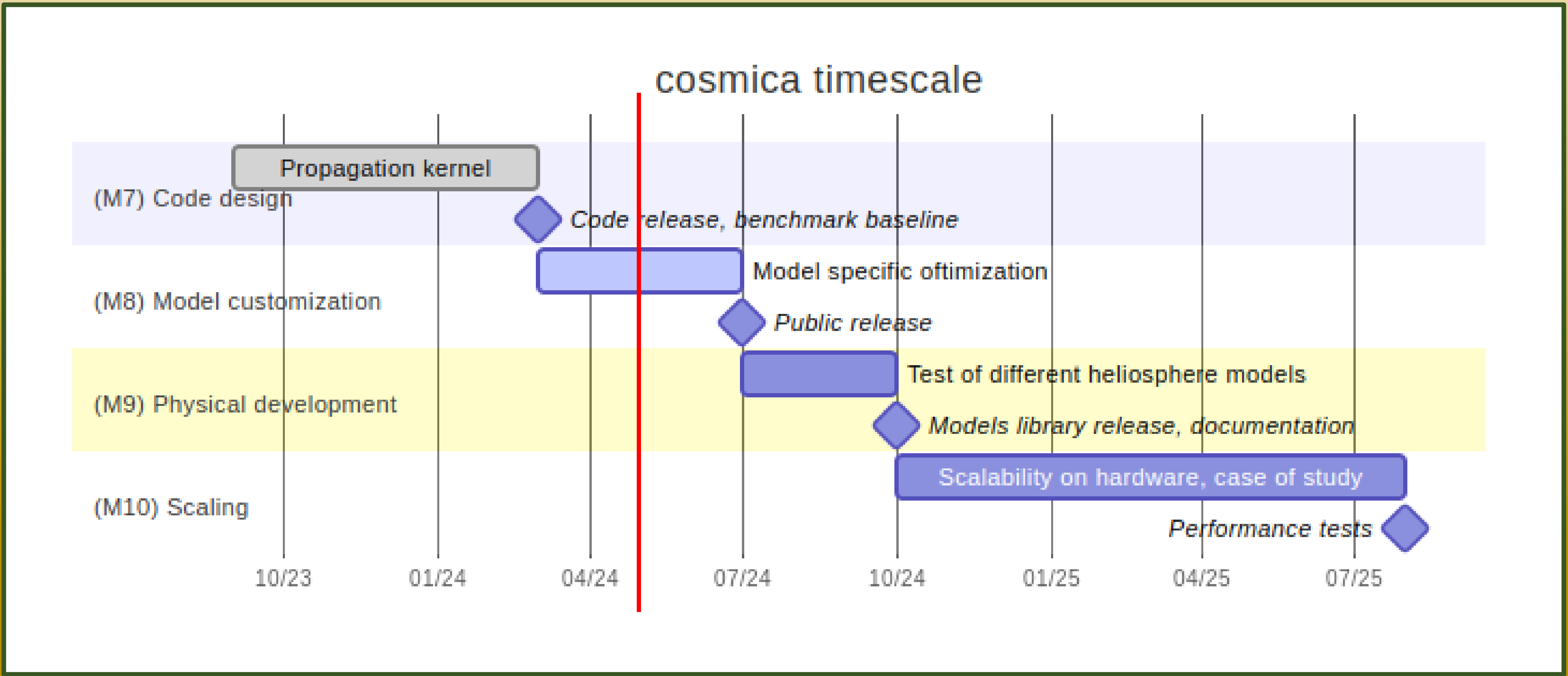
Solutions

Algorithm scheme



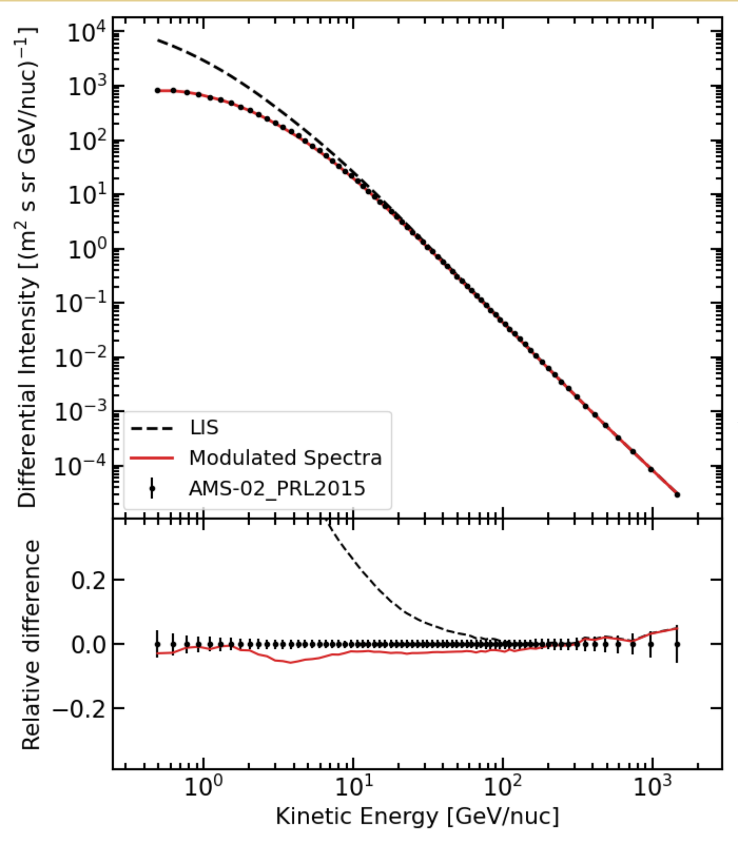
- Manage to distribute the simuated energies on all the available GPUs in the cluster
- Profile the code with Nsight Compute and its execution with Nsight System (bottleneck & GPU usage)
- Achieve the algorithm maximum parallelization by reducing register usage, maximising active threads
- Avoid multiple CPU - GPU memory transfer
- Each warp evolves quasi-particles belonging to the same subset of heliospheric variable (maximize the probability of broadcasting)

Timescale, Milestones, KPIs



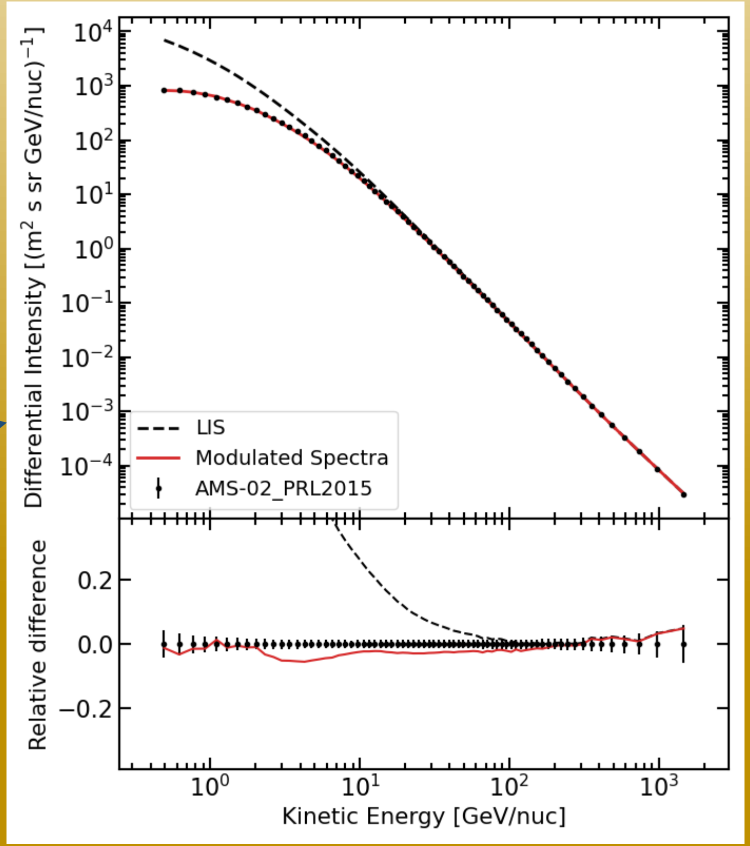
Accomplished Work, Results

- Stable version 1 of Cosmica (within 2% of statistical simulation error)



Cosmica

Preliminary code



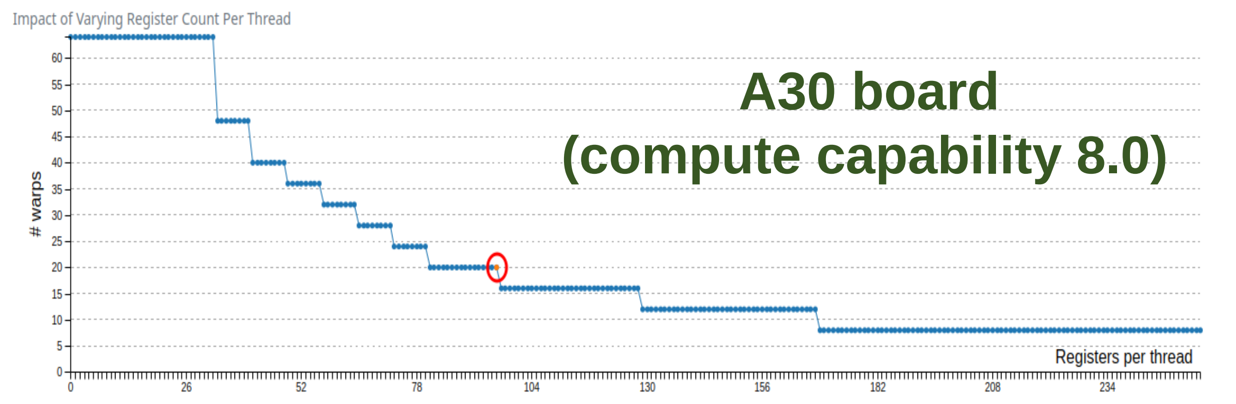
Accomplished Work, Results

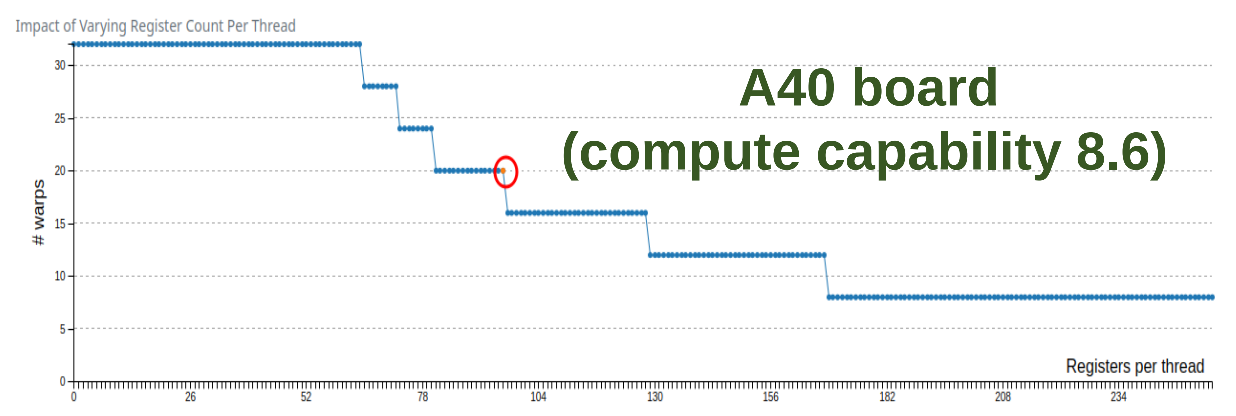
- **Profiling of the execution bottleneck of single GPU kernel functions**

96 registers used for the heliospheric propagation kernel function (maximum registers per thread = 32)

↓

Registers from other warps are allocated and the number of active warps is sub-optimal





Accomplished Work, Results

➢ **Profiling of the execution bottleneck of single GPU kernel functions**

Time	Total Time	Instances	Avg	Med	Min	Max	StdDev	Category	Operation
100.0%	429.333 s	14	30.667 s	27.342 s	15.777 s	57.753 s	12.226 s	CUDA_KERNEL	HeliosphericPropagation(curandStatePhilox4_32_10 *, PropagationParameters_t, particle_t *, i
0.0%	64.642 μs	18	3.591 μs	4.144 μs	608 ns	4.513 μs	1.319 μs	MEMORY_OPER	[CUDA memcpy Host-to-Device]
0.0%	55.904 μs	42	1.331 μs	1.280 μs	1.216 μs	1.792 μs	150 ns	MEMORY_OPER	[CUDA memcpy Device-to-Host]
0.0%	39.616 μs	14	2.829 μs	2.816 μs	2.816 μs	2.848 μs	16 ns	CUDA_KERNEL	kernel_max(particle_t *, float *, unsigned long)
0.0%	28.127 μs	14	2.009 μs	2.016 μs	1.984 μs	2.048 μs	22 ns	CUDA_KERNEL	histogram_atomic(const particle_t *, float, float, int, unsigned long, float *, int *)
0.0%	21.408 μs	14	1.529 μs	1.536 μs	1.504 μs	1.536 μs	13 ns	CUDA_KERNEL	histogram_accum(const float *, int, int, float *)
0.0%	6.560 μs	14	468 ns	480 ns	448 ns	480 ns	15 ns	MEMORY_OPER	[CUDA memset]
0.0%	3.008 μs	1	3.008 μs	3.008 μs	3.008 μs	3.008 μs	0 ns	CUDA_KERNEL	init_rdmgenerator(curandStatePhilox4_32_10 *, unsigned long long)

- Execution time strongly dominated by the heliospheric propagation computation
- Max exit energy search and histogram building are negligible in the execution time
- Even memory set and transfer between host and device occupy less than 0.1%

Accomplished Work, Results

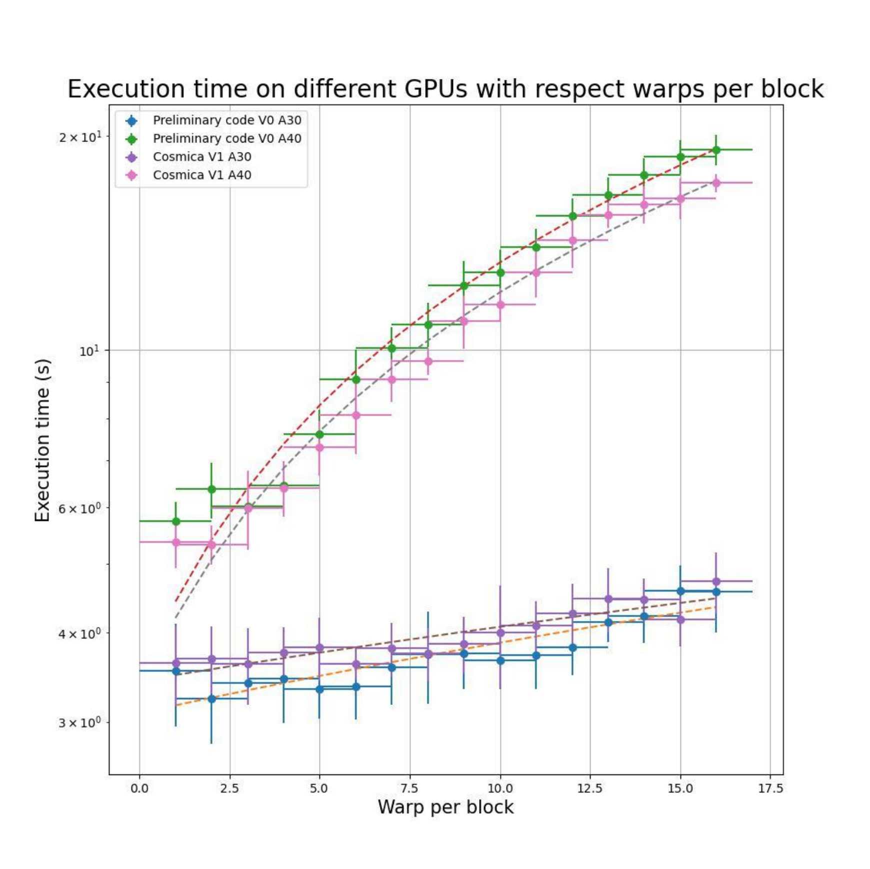
➢ Profiling of the execution bottleneck of single GPU kernel functions

ID	Estimated Speedup	Function Name	De	Duration	Runtime Improvement (1.88419e+11)	Compute Throughput	Memory Throughput	# Registers	Grid Size			Blc	Cycles
0	77.38	init_rdmgenerator	i...	0.00	0.00	0.22	4.75	20	19,	1,	1		5078
1	77.38	init_rdmgenerator	i...	0.00	0.00	0.19	3.97	20	19,	1,	1		6069
2	77.38	init_rdmgenerator	i...	0.00	0.00	0.28	6.07	20	19,	1,	1		3985
3	66.07	init_rdmgenerator	i...	0.00	0.00	0.35	5.30	19	19,	1,	1		4362
4	77.38	init_rdmgenerator	i...	0.00	0.00	0.29	6.27	20	19,	1,	1		3880
5	77.38	HeliosphericPropag...	...	22.33	17.28	8.70	1.87	106	19,	1,	1		29148005356
6	77.38	HeliosphericPropag...	...	25.74	19.92	8.38	1.82	106	19,	1,	1		33595308941
7	66.07	HeliosphericPropag...	...	18.35	12.12	1.61	2.61	108	19,	1,	1		17053851356
8	77.38	HeliosphericPropag...	...	27.41	21.21	7.95	1.76	106	19,	1,	1		35775477854
9	77.38	HeliosphericPropag...	...	33.64	26.03	7.71	1.67	106	19,	1,	1		43896109450
10	97.62	kernel_max	...	0.00	0.00	0.30	1.33	16	2,	1,	1		6114
11	97.62	kernel_max	...	0.00	0.00	0.30	1.34	16	2,	1,	1		6139
12	97.62	kernel_max	...	0.00	0.00	0.30	1.34	16	2,	1,	1		6135
13	97.62	kernel_max	...	0.00	0.00	0.30	1.34	16	2,	1,	1		6083
14	97.62	histogram_atomic	...	0.00	0.00	0.10	2.00	16	2,	1,	1		4902

- The grid for this launch is configured to execute only 19 blocks, which is less than the GPU's 56 multiprocessors, underutilizing some multiprocessors
- Between 66 - 77% improvement of the most time comsuming function

Accomplished Work, Results

➢ Performance test on the local farm and different GPUs comparison



Comparable performance for A30 board

Small improvement for A40 board

NVIDIA A30

- Compute capability: 8.0
- Clock rate: 1 440 000
- Multiprocessor count: 56
- Warps per Multiprocessor: 64

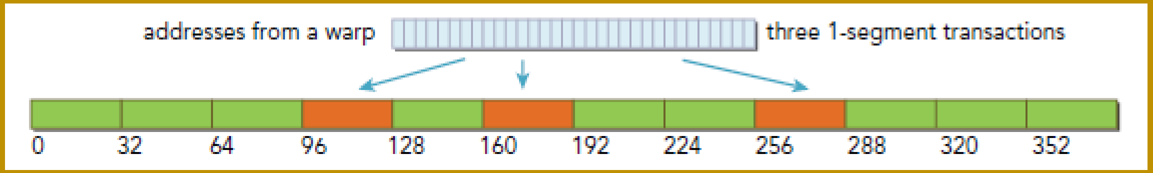
NVIDIA A40

- Compute capability: 8.0
- Clock rate: 1 740 000
- Multiprocessor count: 84
- Warps per Multiprocessor: 48

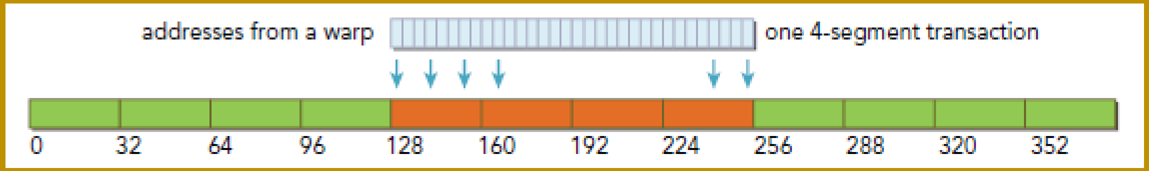
Accomplished Work, Results

- **Preliminary reduction of register usage** → Quasi-particle coordinate and energy are stored in shared memory
- **N threads allocated and warps per block optimization** → Best WpB for each GPU; N quasi-particle rounded; Full filling the first warps
- **Synchronous memeory access to particle evolving variable**

Array of struct



Struct of array



Next Steps and Expected Results

NEXT STEPS:

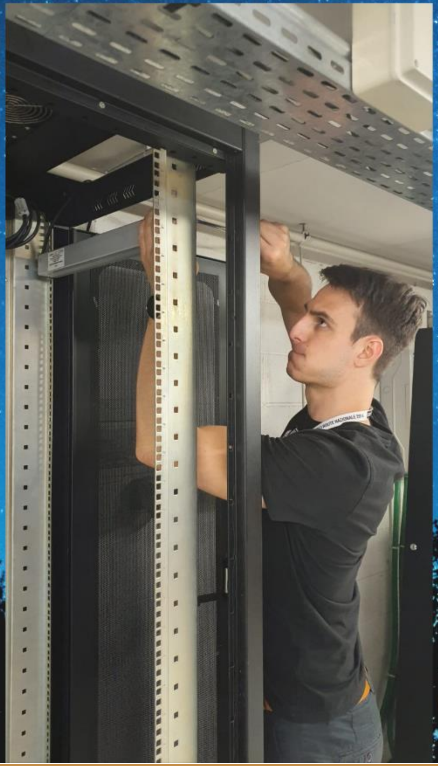
- Passing from particle energy to rigidity (one SDE becomes trivial)
- Customization of the model computations to reduce register usage
- Insert the student thesis work (reduction of 1 model parameter)
- Heliosheat new analysis (paper just accepted)
- Study of K_0 diffusion parameter during Forbush (application for PICA resource)

Expected results:

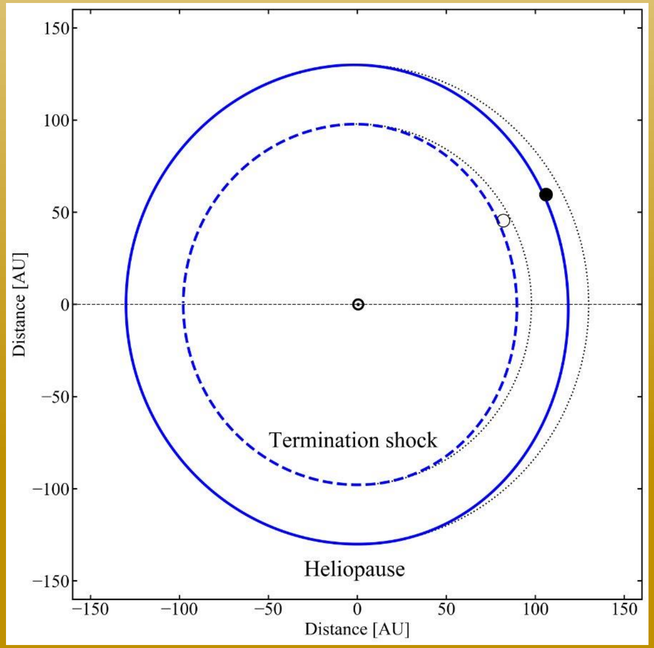
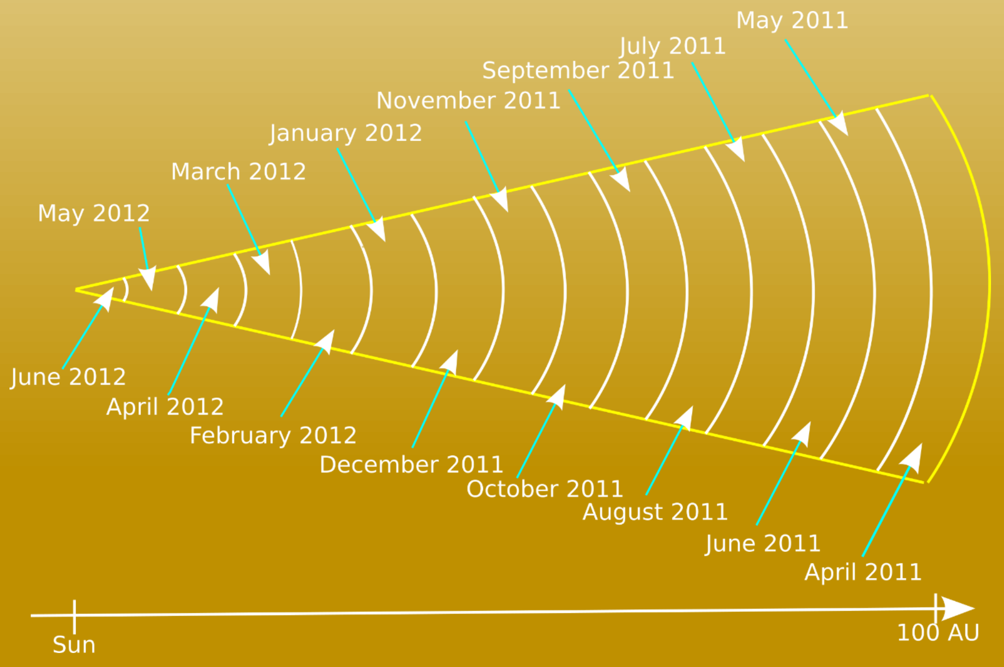
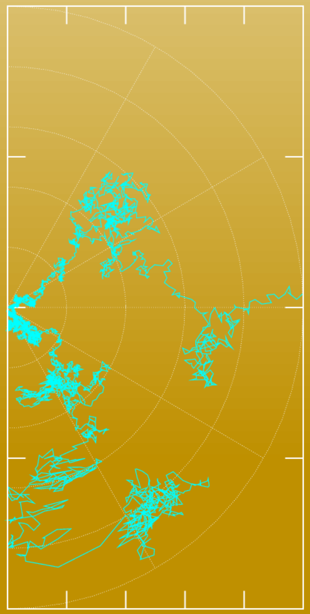
- 3x speed up (exploit the maximum occupancy of GPU)
- Test and compare different physical model numerical stability

Tanks you

For attention

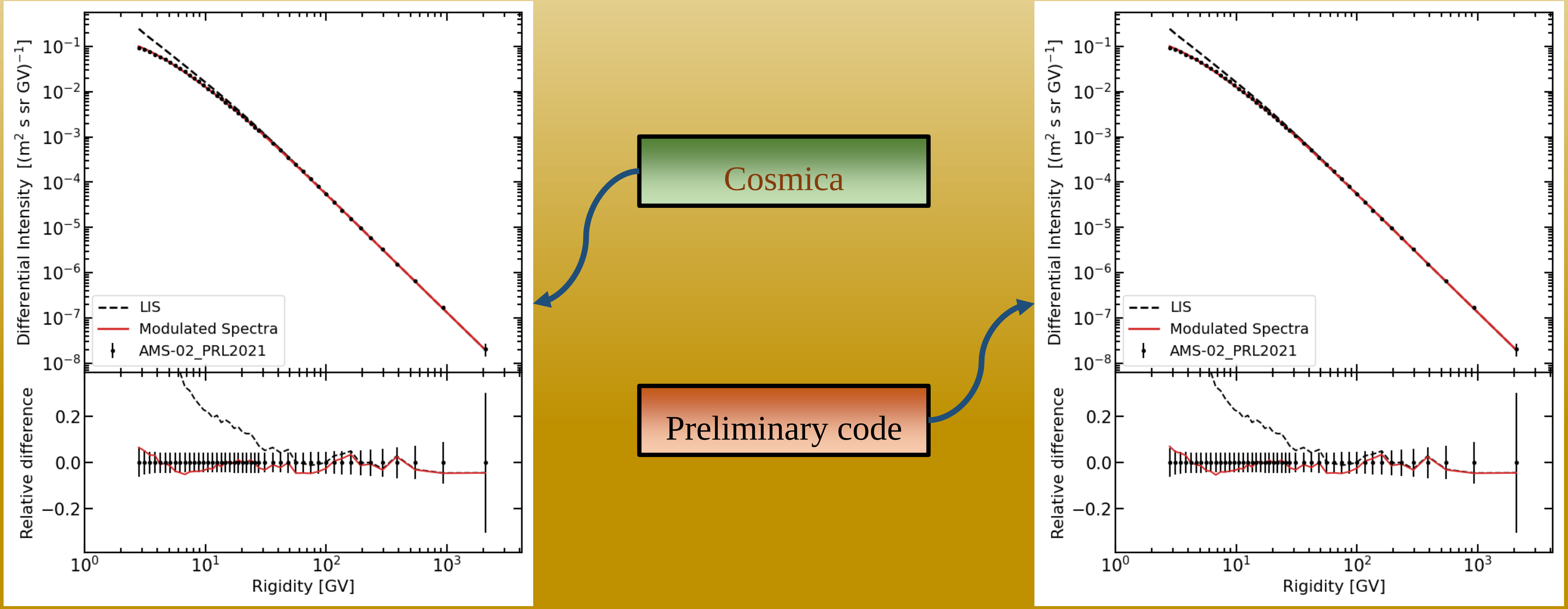


Methodologies

Heliosphere & Heliosheat	Carrington regions	Particle propagation path
		

Accomplished Work, Results

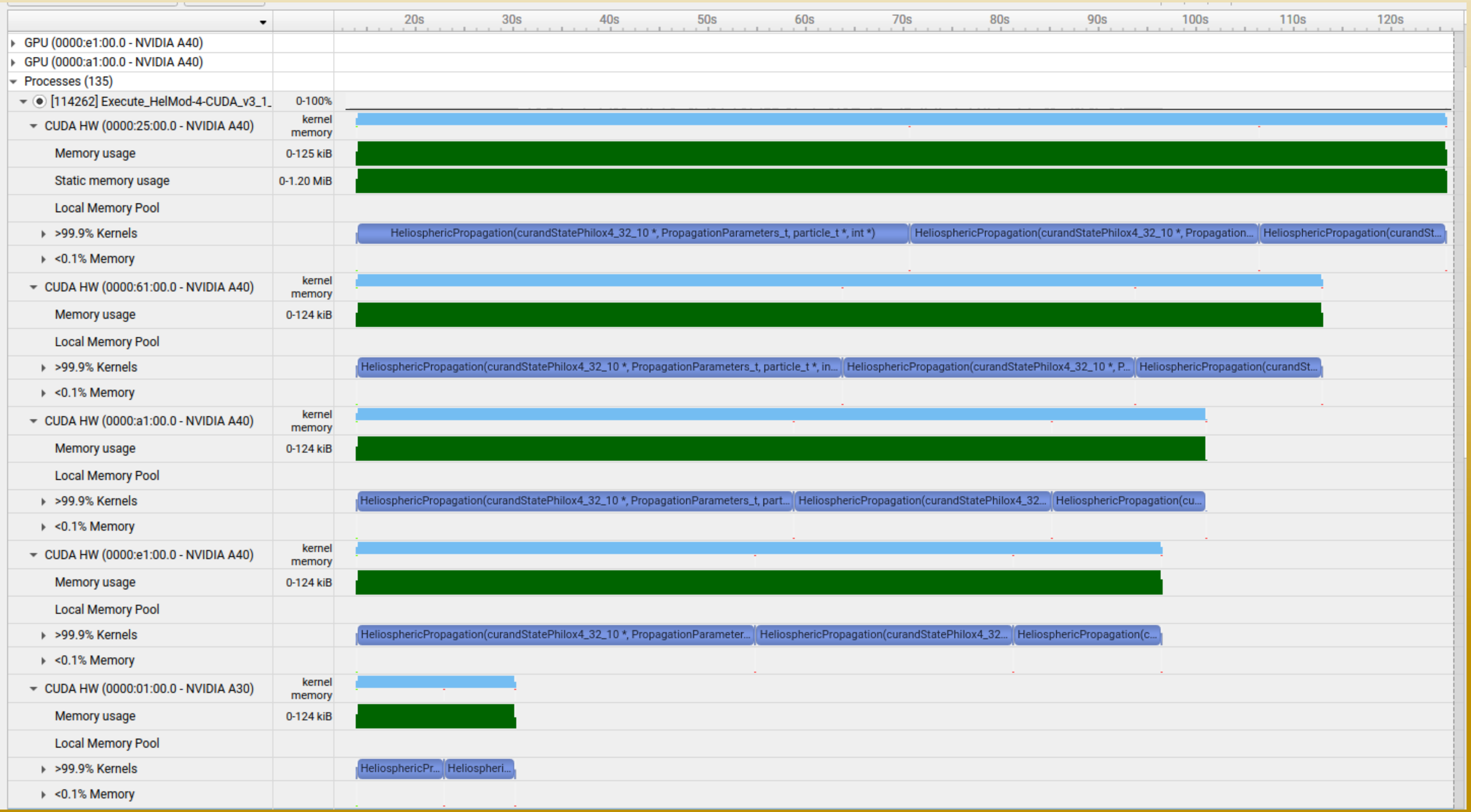
- **Stable version 1 of Cosmica (within 2% of statistical simulation error)**



Nsight analysis (multi energy)

Multi GPu

- Energy bins with different avarage propagation time are not equally distributed between the GPU in the cluster
- Evident faster execution of the propagation computation in the A30 boards



- The number of warps per block was varied from 2 to 32 for each GPU board (only the avilable values of warps per block are taken into account)

Register allocation unit size	256
Register allocation granularity	warp
Max registers per Block	65536
Warp allocation granularity (for register allocation)	4
Registers used by the kernel	106

$$\textbf{Registers allocated} = \textbf{int\_round}\left(\frac{NRegPerKernel * WarpSize}{RegAllocUnit * WarpGranularity}\right) * RegAllocUnit * WarpGranularity * NWarps$$

⬇

With more that 16 warps we exceed the GPU resources

⬇

Nregisters = 73 728 for 18 Warps

We use 106 registers for the heliospheric propagation kernel function (maximum registers per thread = 32)

⬇

Registers from other warps are allocated and the number of active warps is sub-optimal

Finanziato dall'Unione europea
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Nsight compute analysis

A40 multi energy

ID	Estimated Speedup	Function Name	De	Duration	Runtime Improvement (5.04838e+11)	Compute Throughput	Memory Throughput	# Registers	Grid Size	Blc	Cycles
0	92.86	init_rdmgenerator	i...	0.00	0.00	0.29	4.21	19	4, 1, ...		5703
1	95.24	init_rdmgenerator	i...	0.00	0.00	0.21	4.34	20	4, 1, ...		5805
2	95.24	init_rdmgenerator	i...	0.00	0.00	0.20	4.32	20	4, 1, ...		5838
3	95.24	init_rdmgenerator	i...	0.00	0.00	0.20	4.28	20	4, 1, ...		5879
4	95.24	init_rdmgenerator	i...	0.00	0.00	0.20	4.14	20	4, 1, ...		6087
5	92.86	HeliosphericPropag...	...	17.33	16.09	1.62	2.58	108	4, 1, ...		16115947797
6	95.24	HeliosphericPropag...	...	66.38	63.22	3.79	0.82	106	4, 1, ...		86615042692
7	96.43	kernel_max	...	0.00	0.00	0.49	0.79	16	2, 1, ...		5565
8	95.24	HeliosphericPropag...	...	70.58	67.22	3.71	0.81	106	4, 1, ...		92057442068
9	95.24	HeliosphericPropag...	...	52.01	49.53	3.61	0.79	106	4, 1, ...		67881794652
10	95.24	HeliosphericPropag...	...	61.59	58.66	3.56	0.77	106	4, 1, ...		80385971314
11	97.62	kernel_max	...	0.00	0.00	0.30	1.35	16	2, 1, ...		6085
12	97.62	kernel_max	...	0.00	0.00	0.30	1.34	16	2, 1, ...		6126
13	97.62	kernel_max	...	0.00	0.00	0.30	1.32	16	2, 1, ...		6128
14	97.62	kernel_max	...	0.00	0.00	0.30	1.35	16	2, 1, ...		6122

- E

- The grid for this launch is configured to execute only 4 blocks, which is less than the GPU's 84 multiprocessors, underutilizing some multiprocessors
- Number of threads per block not a multiple of the warp dimension (rounded in the next version)
- Between 92 – 95% improvement of the most time comsuming function

Stronger effect of the warp per block on the A40 boards

Accomplished Work, Results

- **Stable version 1 of Cosmica**
- **Profiling of the execution bottleneck of single GPU kernel functions**
- **Performance test on the local farm and different GPUs comparison**
- **Preliminary reduction of register usage**
- **N threads allocated and warps per block optimization**
- **Synchronous memeory access to particle evolving variable**