



Finanziato  
dall'Unione europea  
NextGenerationEU



Ministero  
dell'Università  
e della Ricerca



Italiadomani  
PIANO NAZIONALE  
DI RIPRESA E RESILIENZA



Centro Nazionale di Ricerca in HPC,  
Big Data and Quantum Computing

**The Frequency-Hough project: algorithm  
optimization and HPC for present and future  
continuous gravitational-wave searches.**

**Lorenzo Pierini**

[lorenzo.pierini@roma1.infn.it](mailto:lorenzo.pierini@roma1.infn.it)

Pia Astone  
Marco Serra

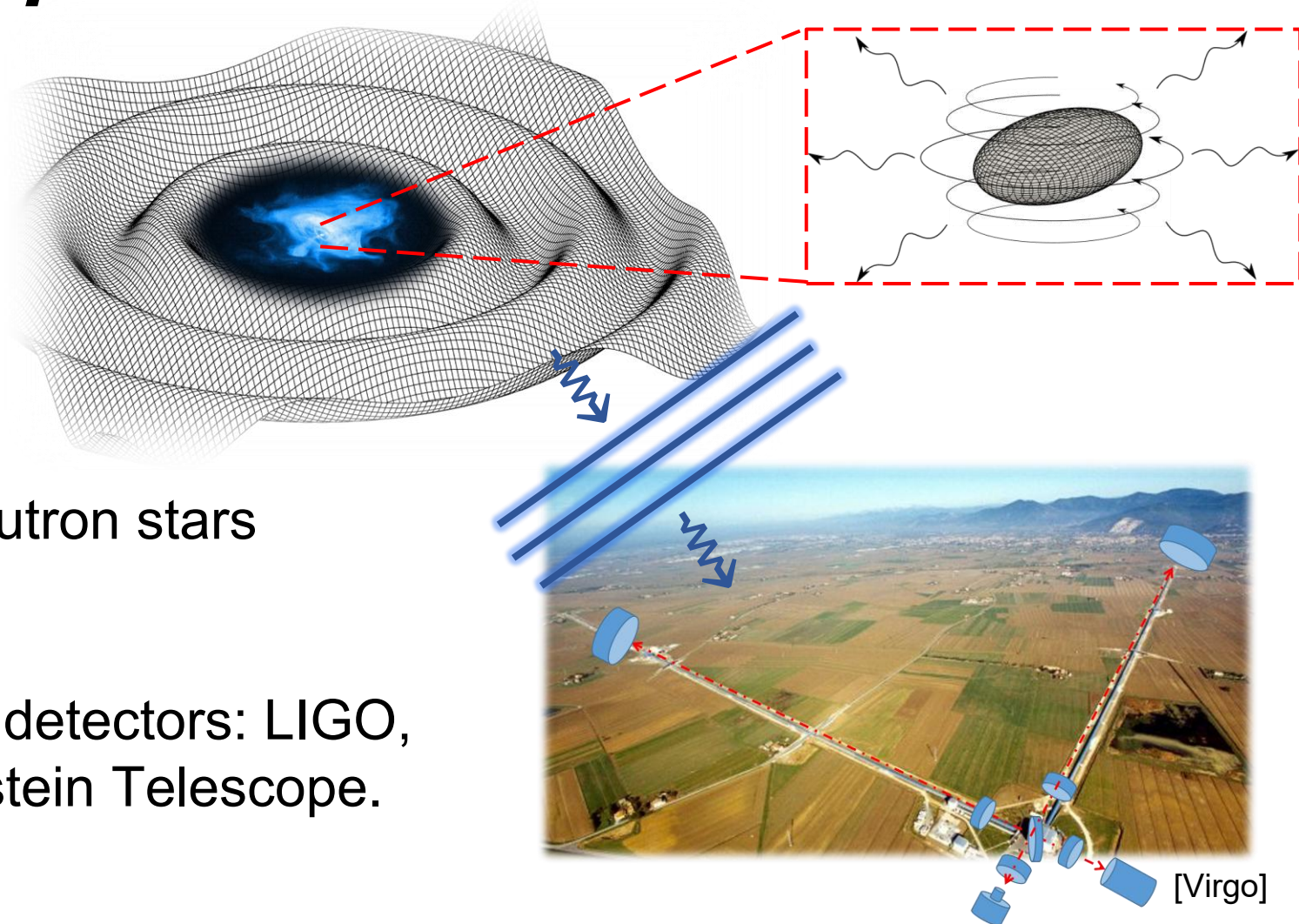
Stefano Dal Pra  
Cristiano Palomba

Spoke 2 annual meeting, Catania, 12/12/2024

# Frequency Hough analysis for continuous gravitational waves.

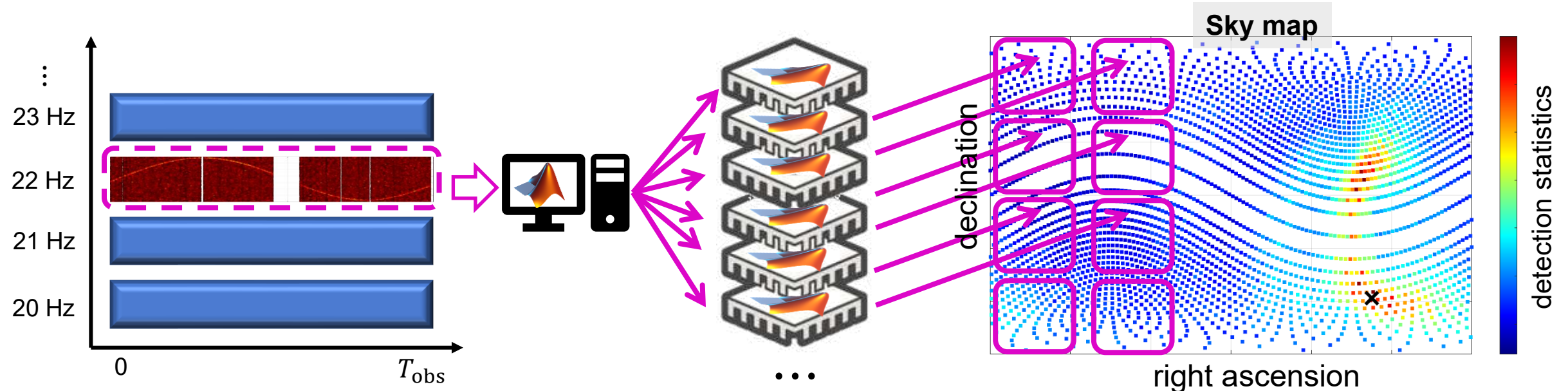
Topic: Long-duration gravitational waves, likely present during the whole data taking.

- Perturbations of the space-time, predicted by General Relativity.
- Emitted by rotating, deformed neutron stars (and more exotic sources).
- High scientific value.
- Can be detected by Earth-based detectors: LIGO, Virgo, KAGRA and by future Einstein Telescope.
- Not yet detected so far.



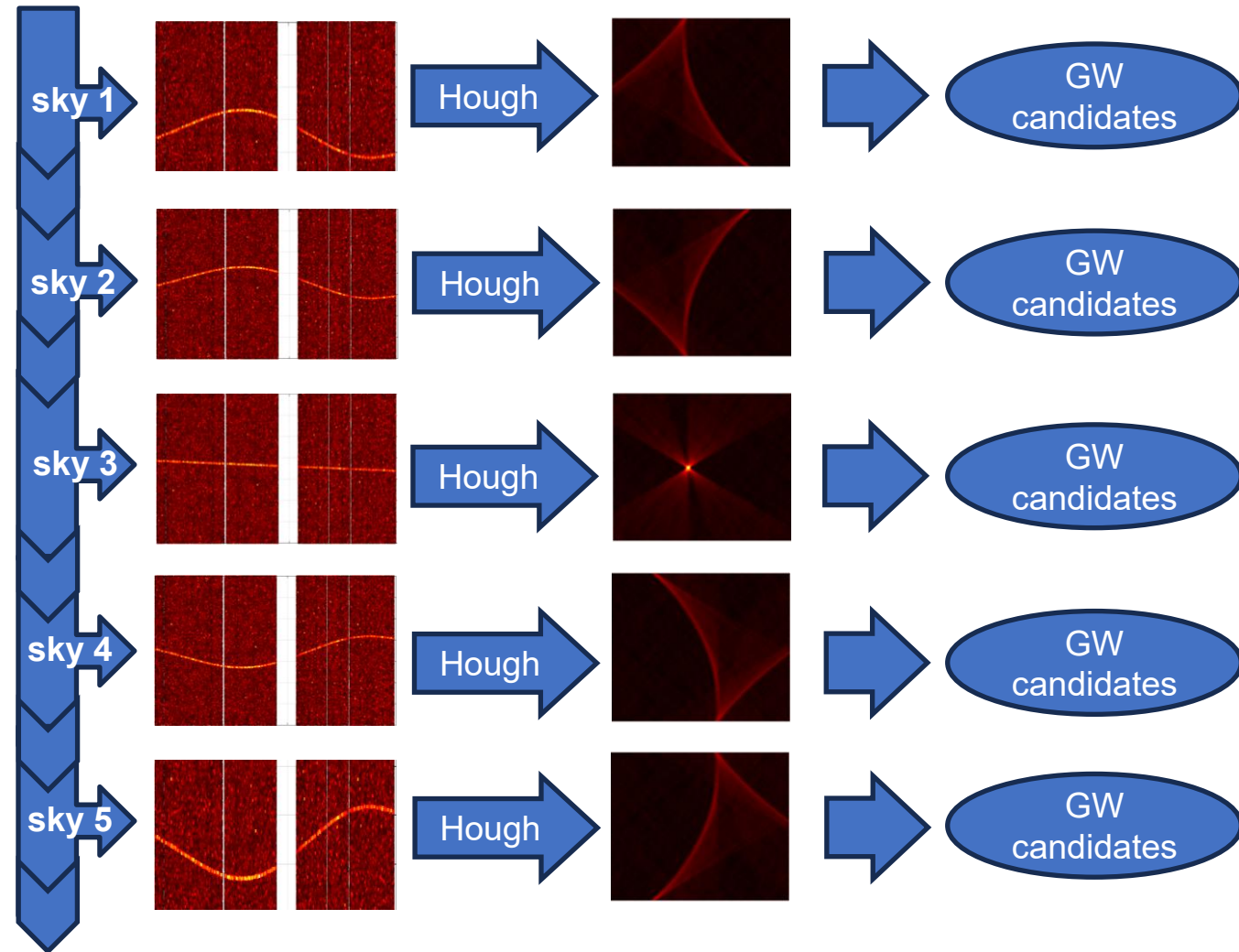
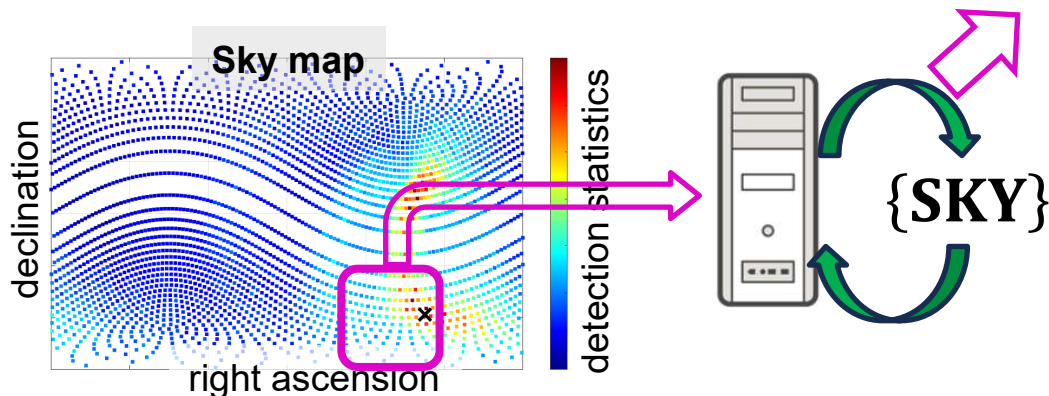
# The algorithm and our project

- 1) Implemented in Matlab.
  - 2) The full search is divided in independent jobs.
  - 3) Each job searches signals in one frequency band (from 20 Hz to 2000 Hz), from sources localized on a subset of directions from a discrete sky map.
  - 4) To complete the full search, we need about  **$10^7$  core hours for 1-year data for each detector!**
  - 5) Using, e.g., **3500** (14HS06) **CPU** cores with no pauses between jobs, no crashes, no waiting queues → **9 months** to analyze data from 2 detectors.
- HPC project at ICSC: resources and optimization.  
→  **$10^7$  core hours** at INFN Grid (Napoli, Bari Tier2)  
→ **400000 GPU hours** at CINECA (Leonardo)



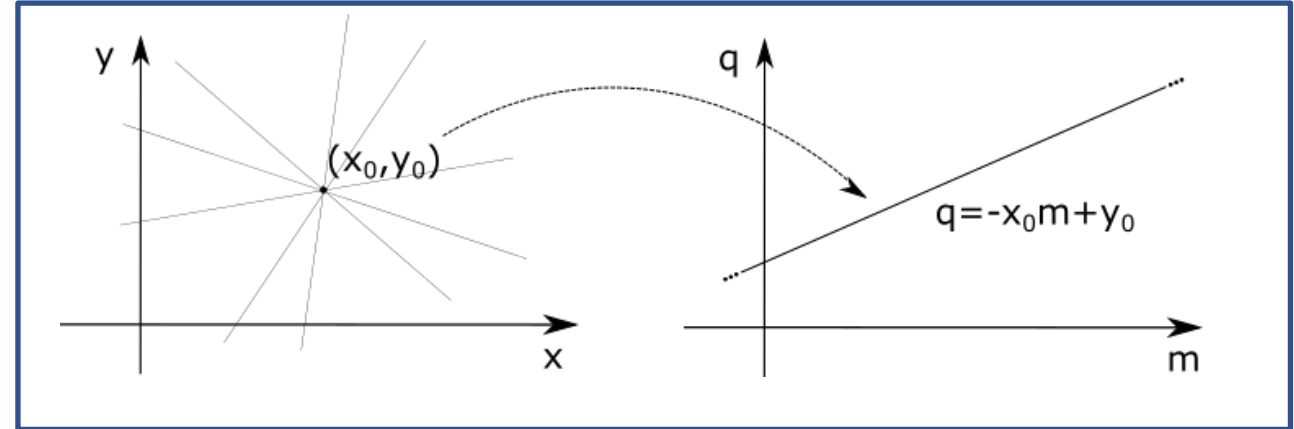
# The core: Hough transform

- The Hough transform is used to detect specific patterns inside an image.
- In our search, time-frequency maps are corrected for a specific sky position and then fed to the Hough transform.
- All the subset of sky positions is analyzed.
- Only the sky position that matches the real position of a source maximizes the output of the Hough transform.
- The Hough transform dominates the total computational cost of the search.

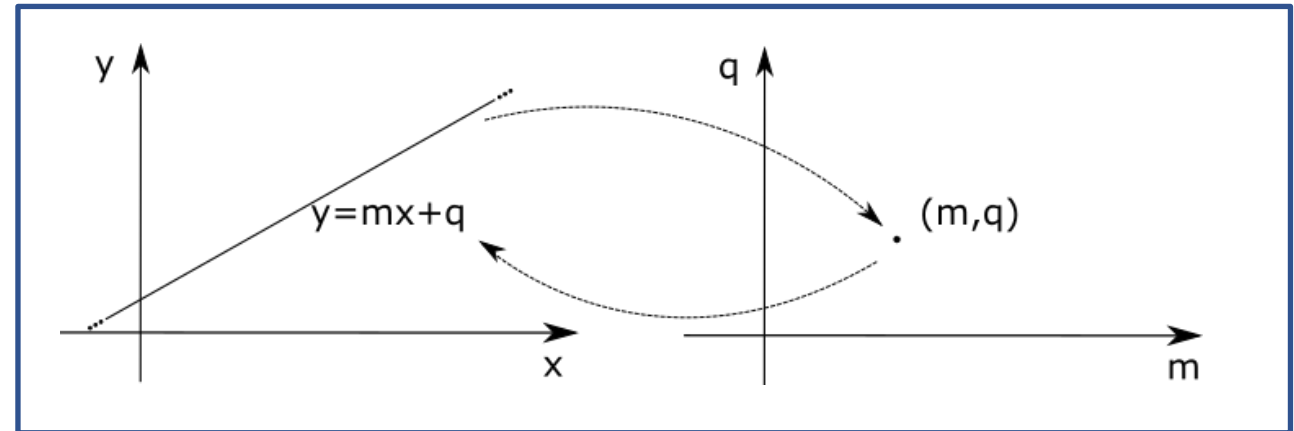


# The optimization of the Hough transform

- ❑ The Hough transform is an integral transform: maps straight lines in the physical space into dots in the parameter space.
- ❑ Old implementation: direct writing. Each point is mapped as a line in the Hough map. The code was highly optimized, with a core compiled in c. But different points can write on the same memory locations → NOT parallelizable!
- ❑ New implementation: reading-writing. Each point in the Hough map is computed by integrating along the corresponding line. → Can be done in parallel!

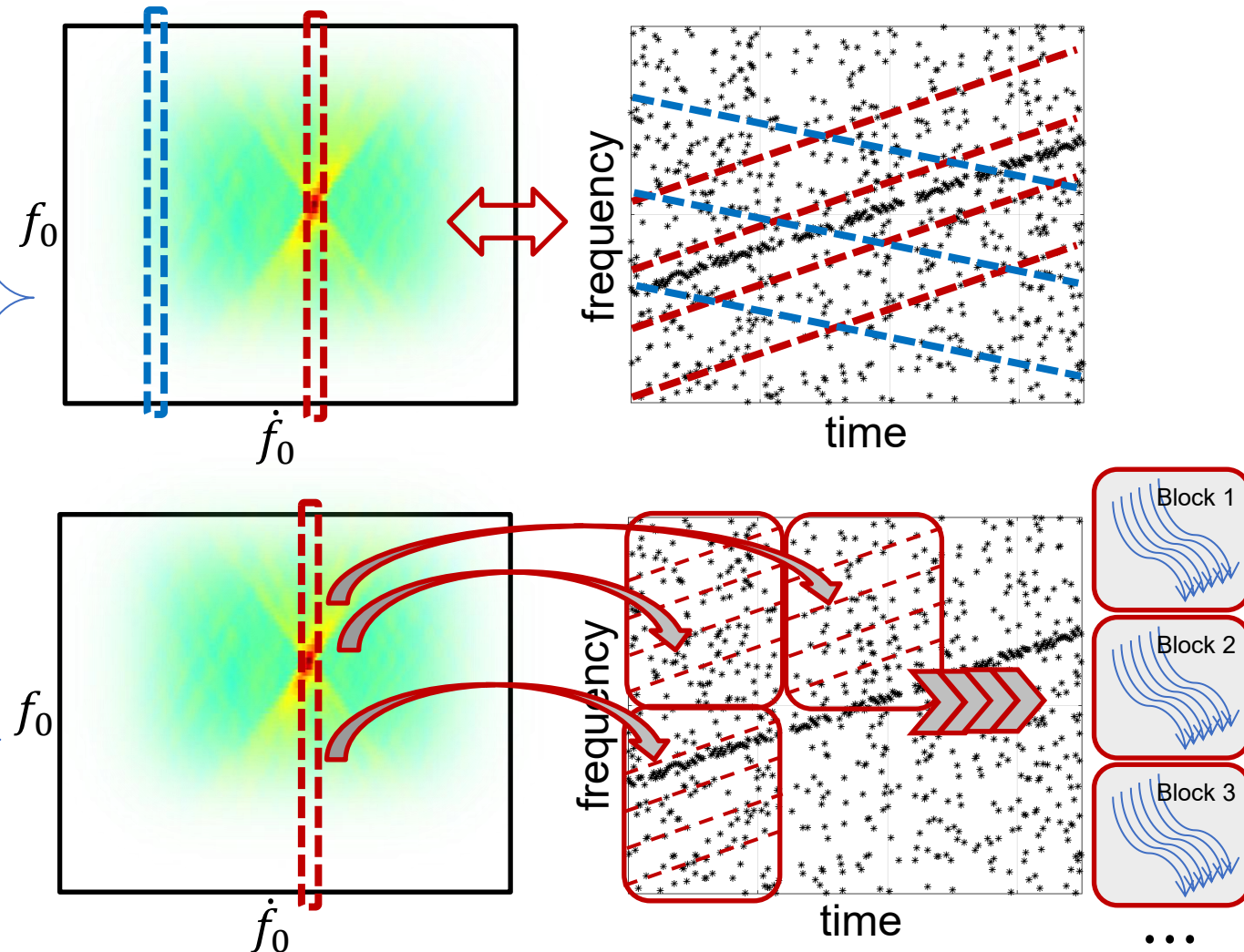


Physical space  $(x, y) \rightarrow y = mx + q \rightarrow$  Parameter space  $(m, q)$



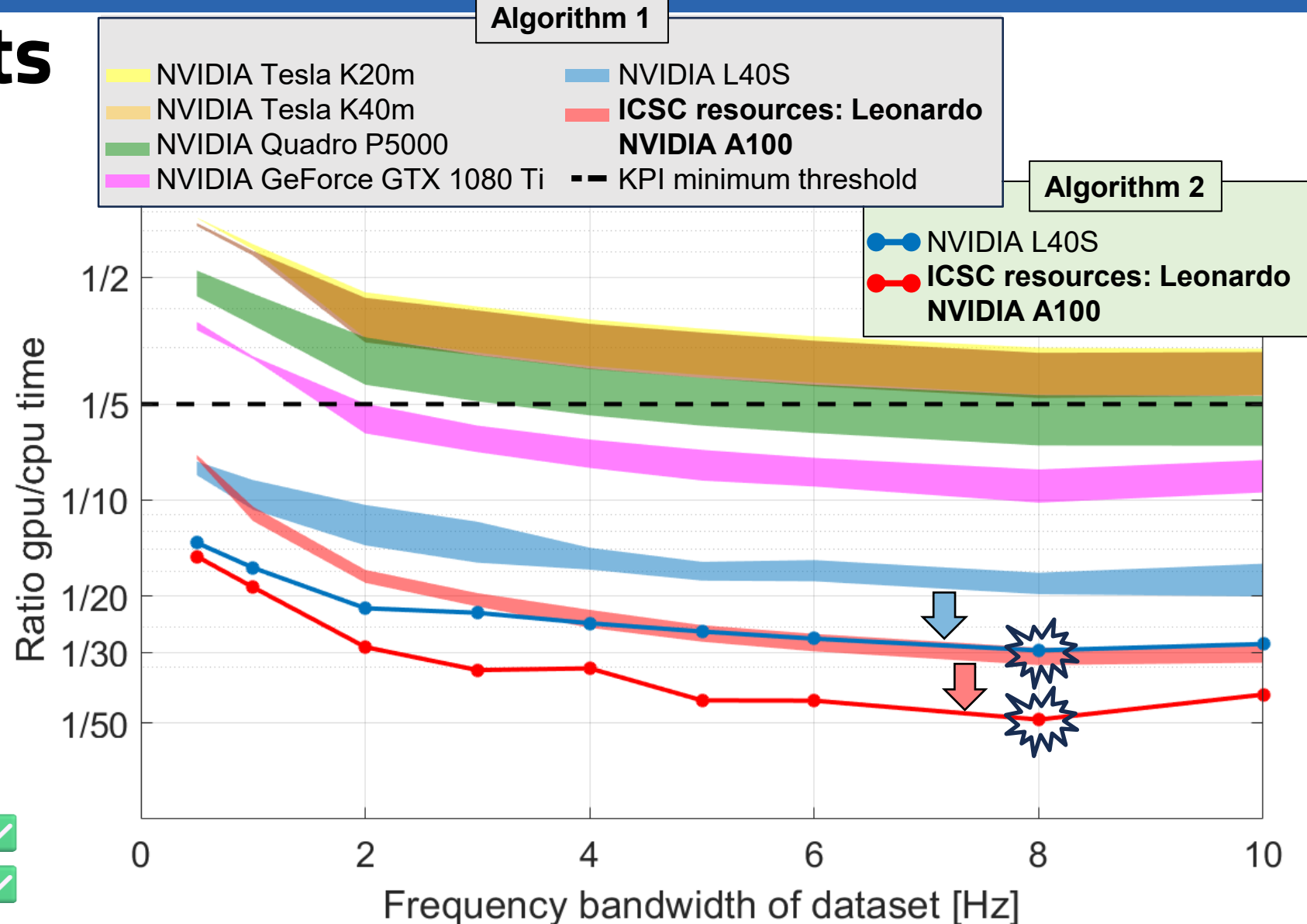
# The workflow through the milestones

- ❑ **MS7**: new implementation on CPU.  
Same performances of the old one when running in single core, but it can exploit hyperthreading.
- ❑ **MS8**: first implementation on GPU  
→ Algorithm 1  
Hough map computed column-by-column, as histograms following different slopes.
- ❑ **MS9**: second implementation on GPU  
→ Algorithm 2  
Compiled CUDA kernel: the whole computation is distributed in blocks, outputs recombined.



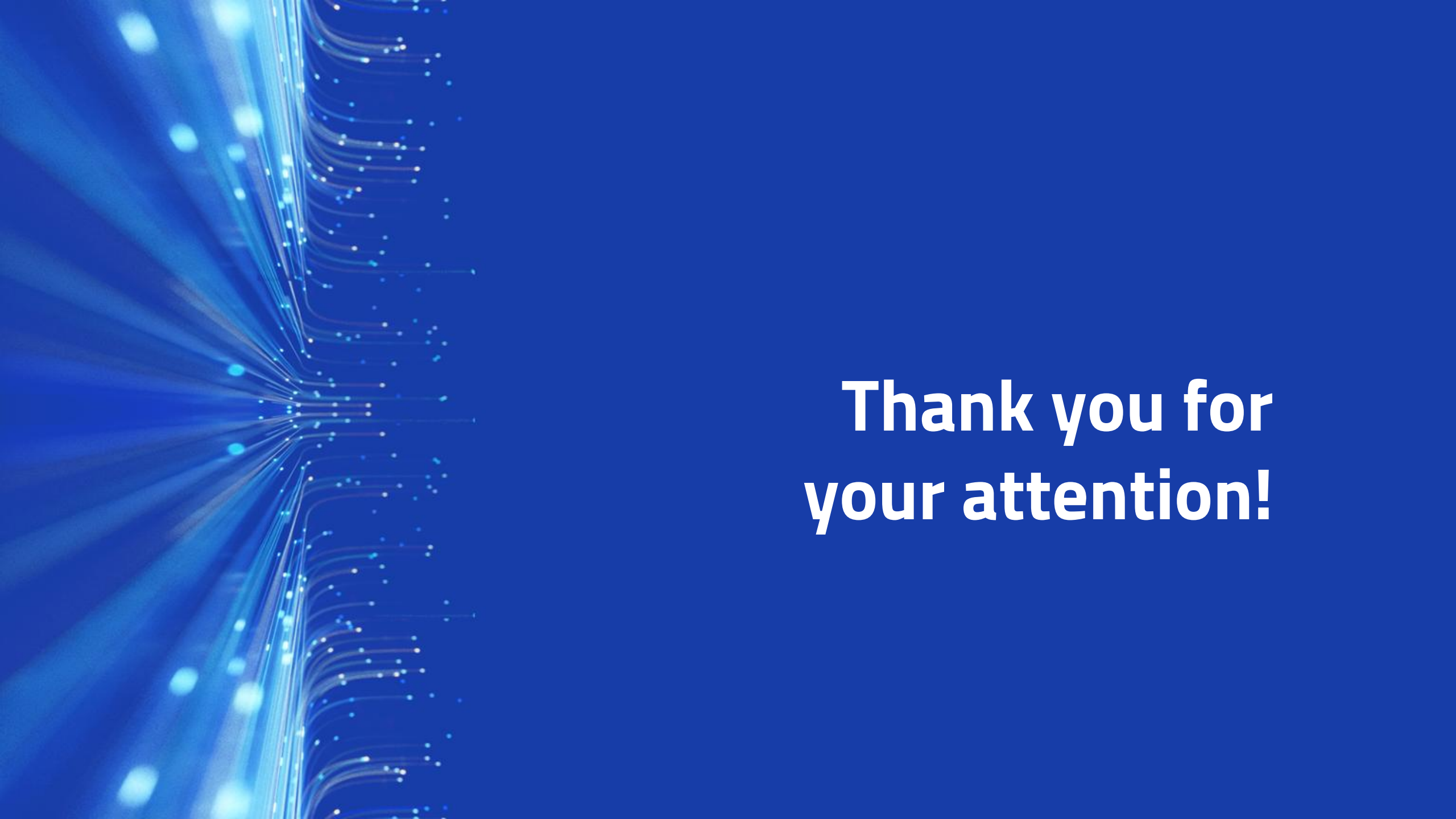
# Performance tests

- For comparison, we run the original code on a recent CPU (Intel Xeon Gold 6430).
- In practice, our gain is higher since in the past we run at CNAF on 11-14 HS06 cores.
- With modern GPUs (L40S, A100) the computing time ratio is significantly below the  $1/5$  threshold.
- With Algorithm 1 the best reduction is by  $1/30$ .
- With Algorithm 2 the best reduction is by  $1/50$ .
- KPI 2.1 – TAR 3.12 fulfilled** ✓
- KPI 2.2 – TAR 3.19 fulfilled** ✓



# Conclusions and next targets

- ❑ Code optimization and HPC are crucial for future gravitational-wave searches.
- ❑ Past milestones (MS 7/8/9): new implementations of the Hough transform, to exploit parallel architectures like hyperthreading and GPUs. The performances of the new algorithms are well beyond the minimal thresholds.
- ❑ Next milestone (MS10): double target.
  - ↪ TAR 3: alternative implementation in python (easier deployment), comparative tests.
  - ↪ TAR 4: optimization of the full pipeline and extensive testing of the computing infrastructure, with long runs and different algorithms.
- ❑ New milestone for ICSC extension? Maybe we have!



**Thank you for  
your attention!**