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

Advancing radioastronomy into the era of High Performance Computing: the case of Imaging

CLAUDIO GHELLER
INAF – INSTITUTE OF RADIOASTRONOMY BOLOGNA

Emanuele De Rubeis (UniBO-IRA), Giuliano Taffoni (OATS), Giovanni Lacopo (OATS), Luca Tornatore (OATS), David Goz (OATS)

RADIOASTRONOMY SOFTWARE

Working at the interface between radioastronomy HPC, my focus is:

- trying to enable the software used for the processing and analysis of radio data to effectively exploit supercomputing solutions,
- address the challenge posed by increasingly larger and complex datasets.
- Providing solutions to improve current codes (NOT to replace them) - WSClean

MOVING TOWARD HPC

HPC currently me

- Parallel-Distril
- Parallel-multit
- Accelerated co
- Memory optim
- Parallel I/O
- Access to Obje
- High-performa
- Integration wit
- Containers
- …. …. …. …. ….

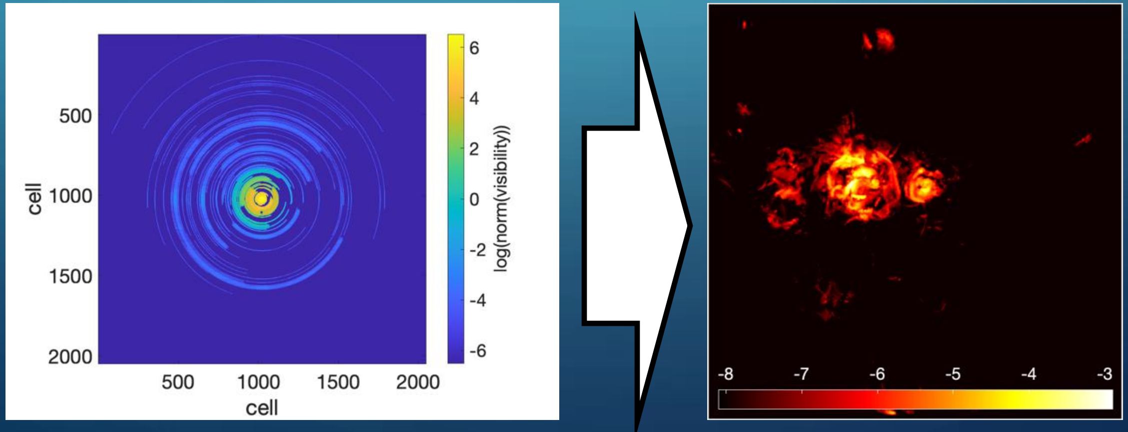
CASE STUDY: IMAGING

Essentially, we want to invert the following integral:

$$V(u,v,w) = \int\int \frac{I(l,m)}{\sqrt{1-l^2-m^2}} \times e^{-2\pi i\left(ul+vm+w\left(\sqrt{1-l^2-m^2}-1\right)\right)} dl\,dm$$

that **maps** the visibilities V measured from the interferometer to the sky brightness I, providing the actual image of the sky.

(u, v, w) are the baselines coordinates, and (l, m) are the sky coordinates.



IMAGING: MAIN STEPS

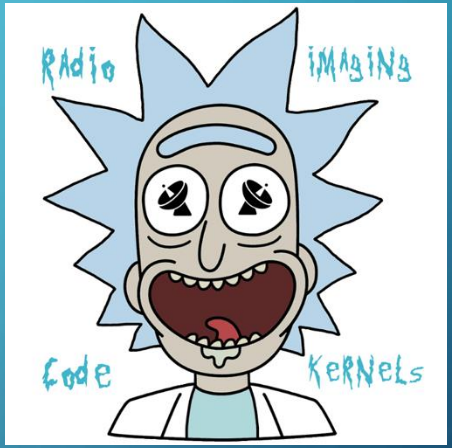
Imaging is a 5D problem: 3 spatial dimensions, frequencies, polarisations.
It requires essentially 3 **computational demanding** operations:

1. **Discretization** of the problem → map visibilities on a regular mesh (needed for FFT) + weighting + tapering
2. **FFT** transform from Fourier to Real space
3. (**W–correction**, if needed, to correct for Earth curvature)

These operations represent computational challenges that can benefit from High Performance Computing.

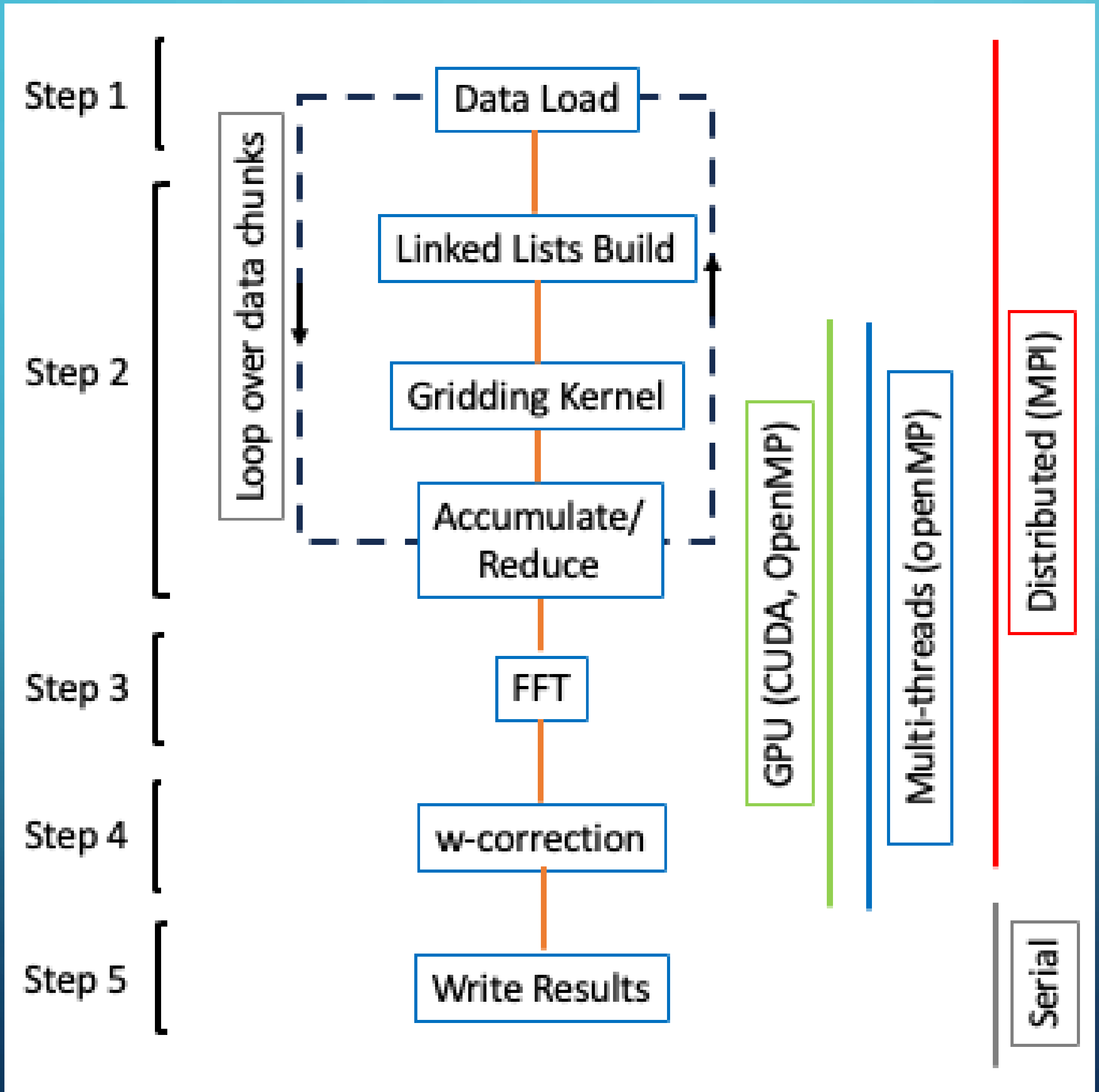
RICK (Radio Imaging Code Kernels)

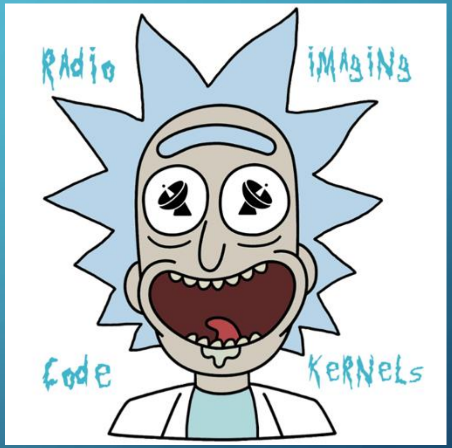
- RICK (Radio Imaging Code Kernels) is a code that addresses the **gridding**, **FFT** and **w-correction**, combining parallel and accelerated solutions.
- It is being designed not to substitute radioastronomy codes but to provide specific solutions, portable and fast
- **C**, **C++**, **CUDA, HIP** (for AMD GPUs)
- **MPI** & **OpenMP** parallel, fully working in parallel
- It can run on GPUs (both CUDA and OpenMP for GPU offloading), in particular the FFT using the distributed CUDA library **cuFFTMp**
- An optimized version of the reduce has been developed on both CPU (combining MPI+OpenMP) and GPU (using NCCL or RCCL, for Nvidia and AMD respectively)



RICK HPC architecture

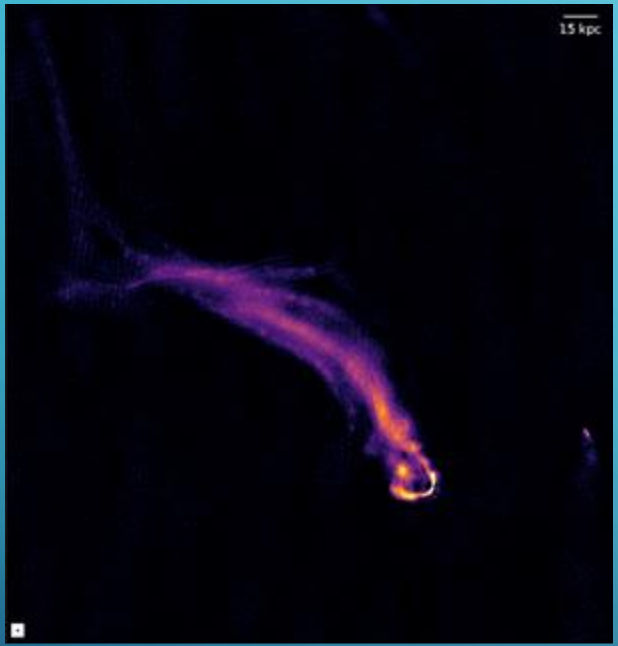
The code is publicly available at
https://www.ict.inaf.it/gitlab/claudio.gheller/hpc_imaging





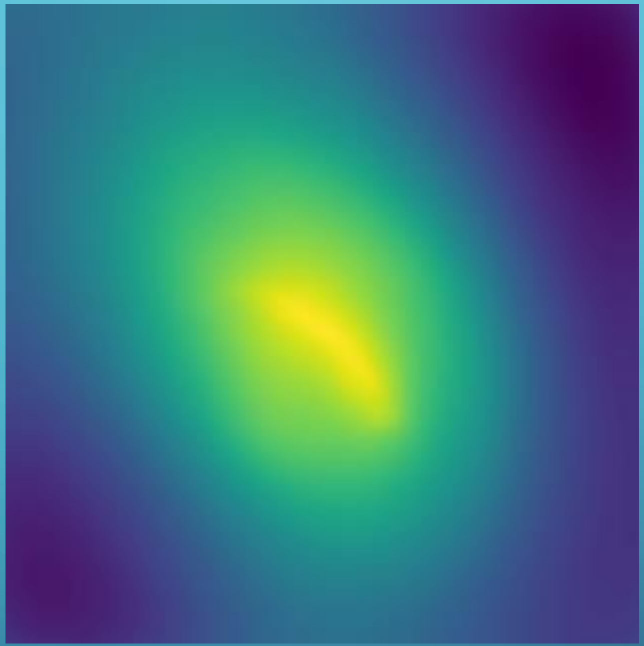
RICK: DOES IT WORK?

Deconvolved image

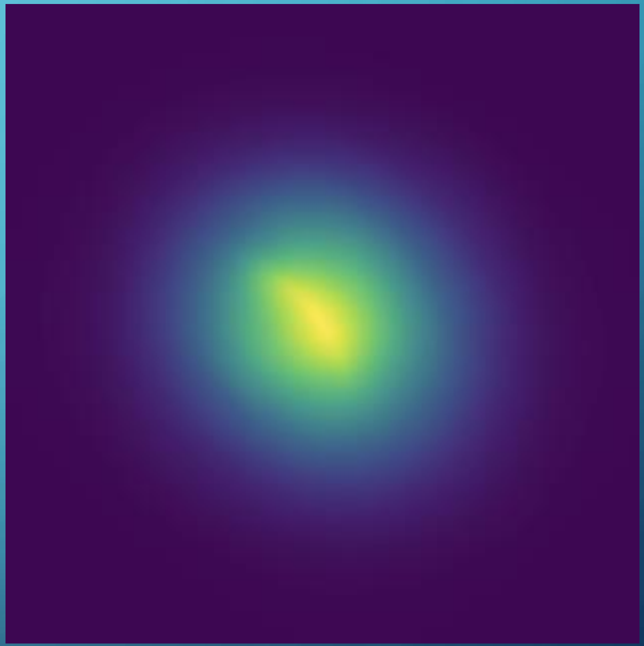


The dataset refers to the "Original TRG", a head-tail radio galaxy in the galaxy cluster Abell 2255

Dirty image, weighting natural, size: 4096x4096x100



WSClean



RICK

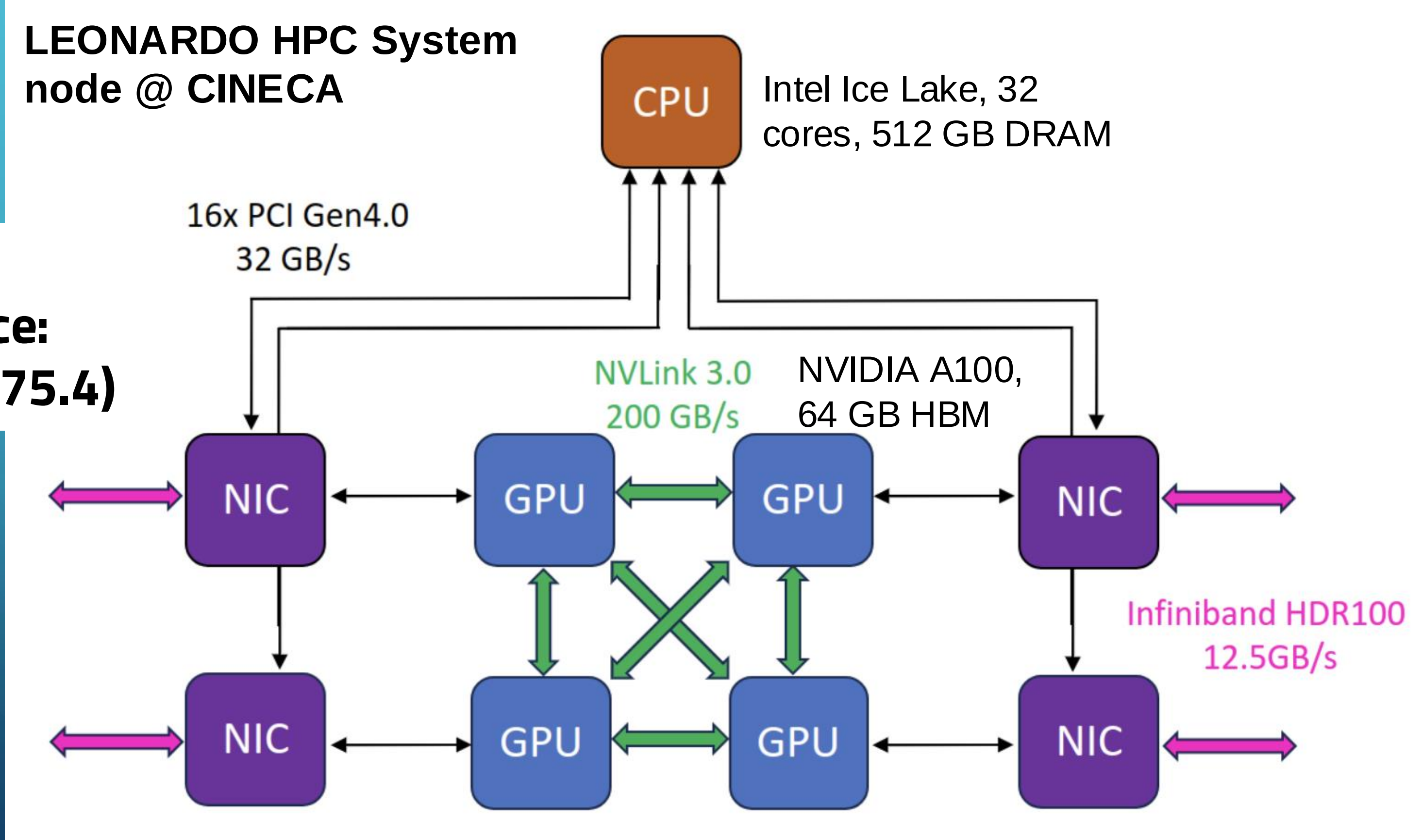
THE COMPUTING SYSTEM: LEONARDO@CINECA

- **4992 Nodes**
- **~300 PFlops**

Showing only a few results. For more, see:

- **De Rubeis at al, Astronomy and Computing, 2024**
- **Lacopo et al, submitted**
- **Gheller et al, SPIE proc., 2024**
- **Gheller et al., RASTI, 2023**

Nominal Performance: 78 TF (2.6+75.4)



Turisini et al, 2024

SINGLE NODE TEST

LOFAR HBA Inner Station:

- 1891 baselines
- 146 MHz, 4 polarizations
- 8 hrs

Data:

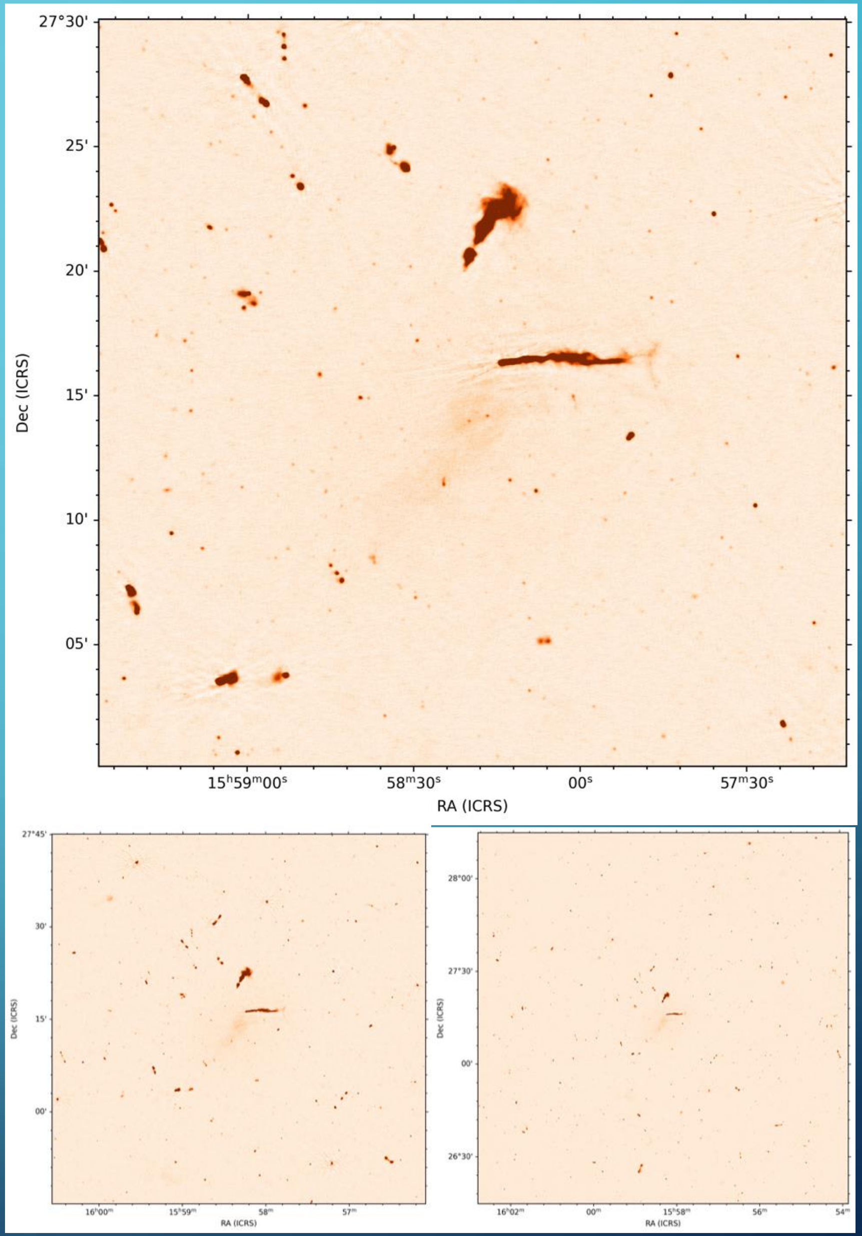
- 543 million visibilities
- 4.5 GB

Mesh/Image:

- 4096x4096 px, 80 MB
- 4096x4096x16, 430 MB

Memory Usage

- ~ 10 GB



Curtesy of Luca Bruno

SINGLE CORE (single node test)

Our BASELINE

Threads	GPUs	Gridding [sec]	FFT [sec]	w-stack [sec]	Comm. [sec]	Total [sec]
1	0	322.06± 2.95	30.01±0.54	26.08± 0.03	0	418.07± 3.25

MULTICORE (single node test)

OpenMP Multithreading allows to use all the cores of a CPU, accessing all its memory

Threads	GPUs	Gridding [sec]	FFT [sec]	w-stack [sec]	Comm. [sec]	Total [sec]
1	0	322.06± 2.95	30.01±0.54	26.08± 0.03	0	418.07± 3.25
32	0	12.26± 0.14	3.07±0.07	2.18± 0.19	0	48.30± 0.80

29x 10x 12x 9x

GPU ACCELERATION (single node test)

Comparing FULL CPU with FULL GPU

Threads	GPUs	Gridding [sec]	FFT [sec]	w-stack [sec]	Comm. [sec]	Total [sec]
1	0	322.06± 2.95	30.01±0.54	26.08± 0.03	0	418.07± 3.25
32	0	12.26± 0.14	3.07±0.07	2.18± 0.19	0	48.30± 0.80
1	1	23.29± 0.02	0.70±0.01	0.1429± 0.0002	0	68.98± 0.61
1	2	9.51± 0.07	0.86±0.10	0.0781± 0.0030	1.80±0.01	47.34± 1.05
1	4	4.53± 0.05	0.53±0.01	0.0403± 0.0008	1.51±0.22	23.67± 0.44

2.7x | 6x | 55x | COMM. | 2x

LOFAR-VLBI USE CASE

LOFAR-VLBI represent a good benchmark to SKA in terms of data volume, with ~15 TB of end-products for a single, 8hrs, observation.

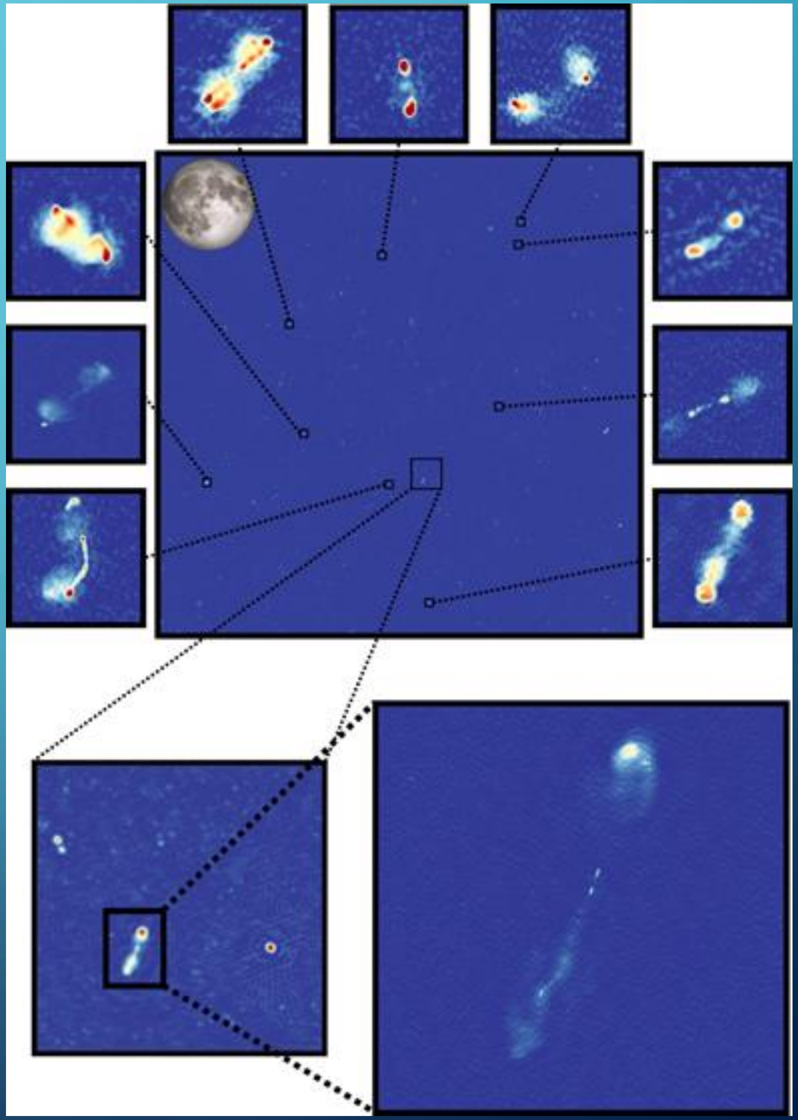
An interesting and challenging task is the imaging of a large FoV (~2.5deg x 2.5deg) with sub-arcsecond resolution using LOFAR-VLBI.

For our LARGE tests, we have:

Input data ~ 533 GB

Output Image = 65536 x 65536 x 32 pixels ~ 4 TB

This is not something a single CPU/GPU can face!

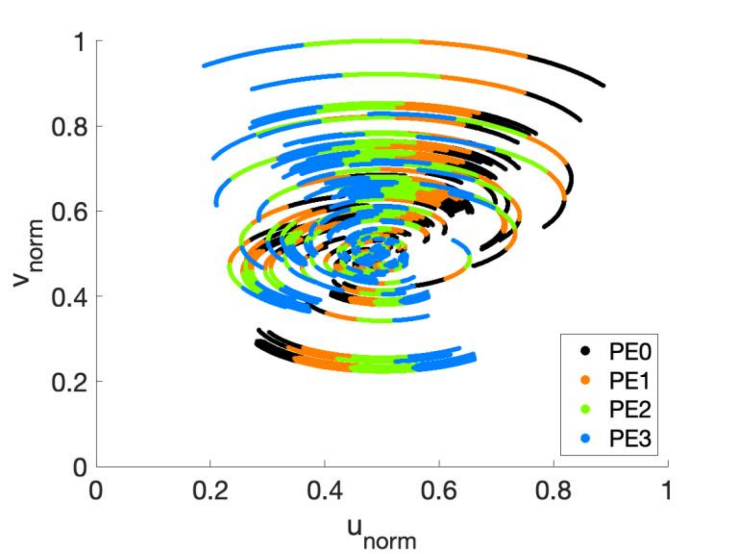


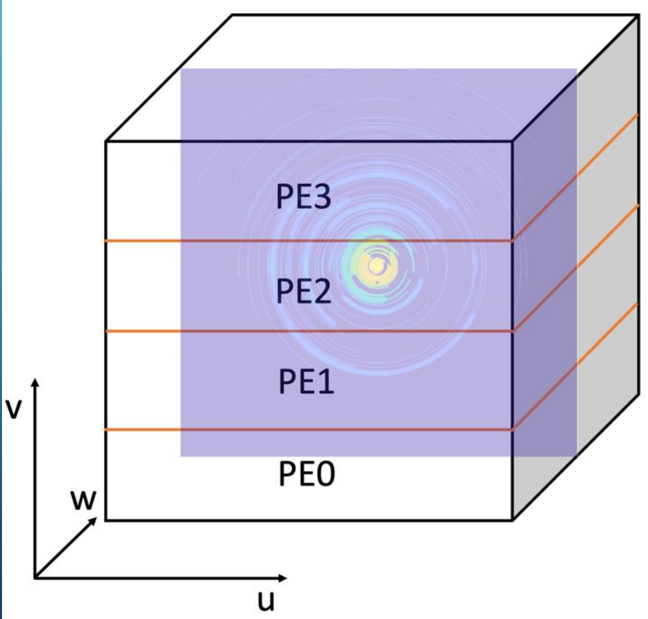
OVERCOMING THE MEMORY WALL

Parallel computing allows to use multiple processors distributing data among their memory:

- Visibilities (and work) are evenly distributed among processing units
- The mesh is split among processing units. The full mesh is never stored in a single memory

→ Problems of "any" size can be supported





Main issue:

visibilities are distributed across memories unrelated to mesh slabs

→ Lots of communication required

LEONARDO@CINECA LARGE TESTS

	Nodes	MPI tasks (threads per task)	GPUs	Gridding (s)	Reduce (s)	FFT (s)	w-correction (s)	Total (s)
CPU tests	32	1024 (1)	0	4.5	9631.4	160.6	7.2	10246.0
	64	2048 (1)	0	1.9	9598.2	107.1	3.5	10153.5
	128	4096 (1)	0	1.1	9715.8	98.4	1.7	10266.5
GPU tests	32	128 (8)	128	2.6	54.8	4.2	0.3	67.4
	64	256 (8)	256	2.4	59.4	2.8	0.2	69.4
	128	512 (8)	512	2.7	72.6	2.7	0.1	83.4

- **The right balance of computing power and scaling must be determined to be efficient**
- **COMMUNICATION BECOMES THE BOTTLENECK**
- **GPUs help reducing the communication bottleneck**

WORK IN PROGRESS & CONCLUSIONS

- **Accurate Evaluation of the Energy impact of RICK**
- **Modularization toward library to be used from WSClean**
- **Optimization of I/O through parallelism**
- **Integration in complex pipelines (Streamflow - https://streamflow.di.unito.it/**

- **HPC can help in strongly reducing the time to solution of radioastronomy data processing of order of magnitudes.**
- **MPI Parallelism is the only**
alleviate I/O bottlenecks
- **Computing nodes must be**
algorithmic components. G ... elp scalability

THANKS FOR YOUR ATTENTION