

E. Michel¹, O. Renaud¹, K. Desnos¹, A. Deller², C. Phillips³, J.-F. Nezan¹

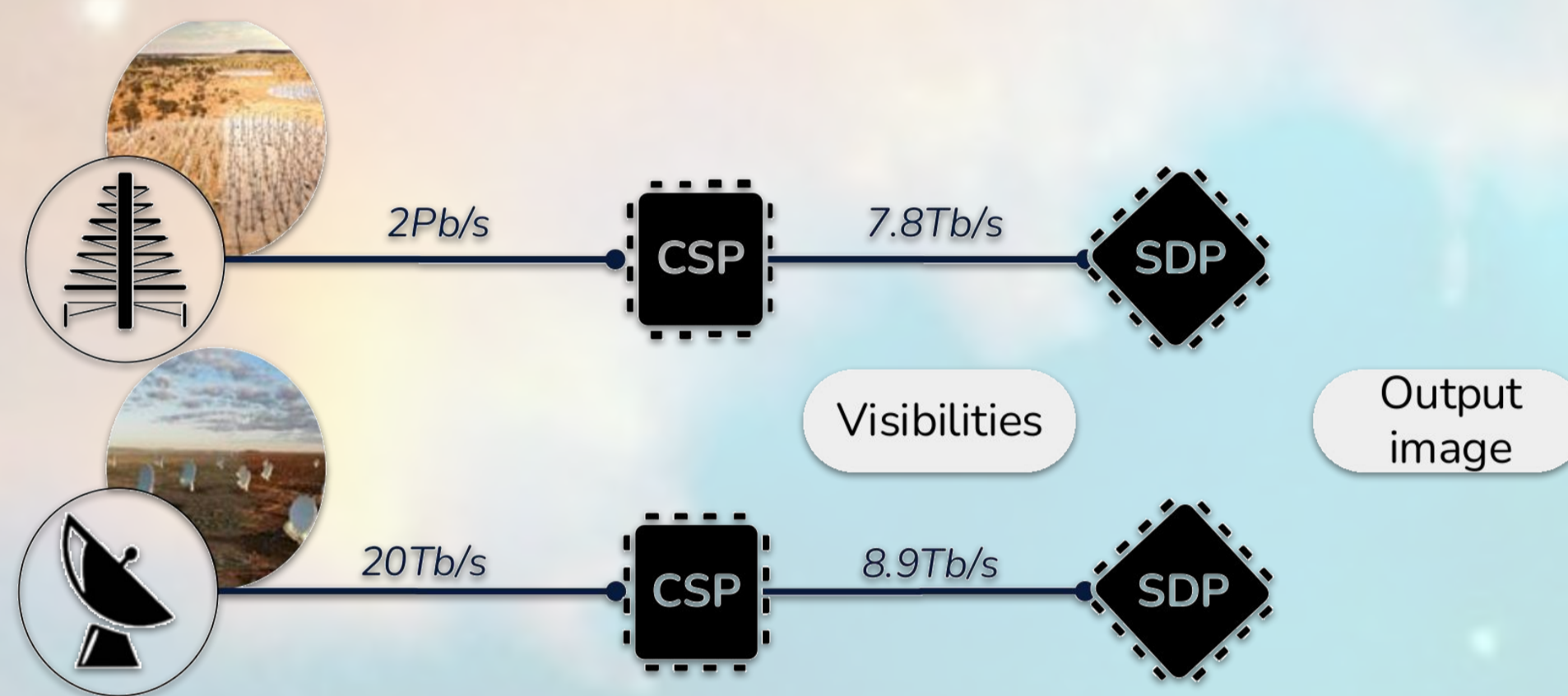
¹Univ Rennes, INSA Rennes, CNRS, IETR - surname.name@insa-rennes.fr

²CAS, Swinburne University of Technology, Hawthorn, Australia - adeller@swin.edu.au

³CSIRO, Space and Astronomy, PO Box 76, Epping, NSW 1710, Australia - Chris.Phillips@csiro.au

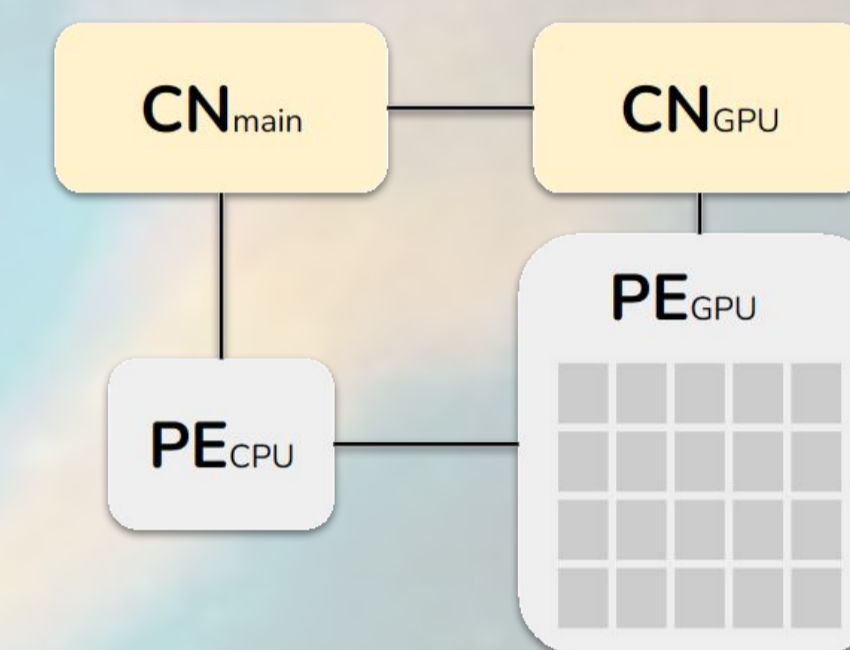
This work was supported by DARK-ERA (ANR-20-CE46-0001-01) and has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 873120. 

The Square Kilometre Array (SKA) exascale radio telescope



- The **Science Data Processor (SDP)** will process data at terabyte rates with a 1 MW, 250 Pflops budget and use **modern HPC system**.
- Resource allocation for complex algorithms and systems is **NP-complete problems**.

System-level architecture modeling for CPU-GPU architecture

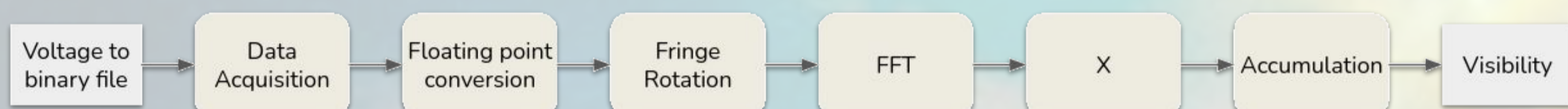


- GPU-oriented System-Level Architecture Model (GSLA):

$$\Lambda = \langle P, C, L, \hat{c}, \lambda \rangle$$

where P is the set of Processing Elements (PEs), C is the set of Communication Nodes (CNs), L is a set of undirected links connecting PEs and CNs, \hat{c} is a property function associating a cost to different elements in the model, and λ is a Lagrangian coefficient setting the cost of a single communication quantum relative to the cost of a single processing quantum.
- Allow **system level** and **internal parallelism** description

DiFX correlator dataflow model



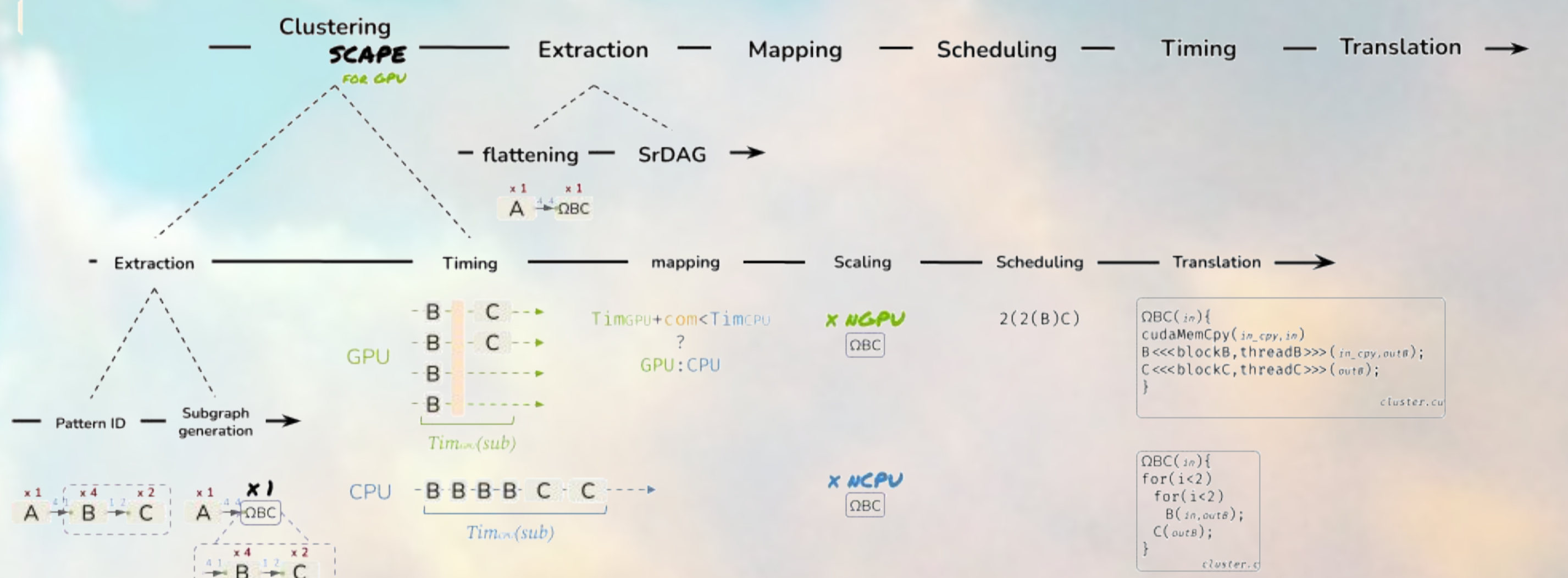
- Six Steps of the DiFX Correlator:
 1. **Data Acquisition**: Extracts sampled/quantized voltage from input (antenna data).
 2. **Floating Point Conversion**: Converts integers to complex numbers (sample + phase shift).
 3. **Fringe Rotation**: Corrects Doppler shifts caused by Earth's rotation.
 4. **FFT (Fast Fourier Transform)**: Separates signal frequencies (noise vs. useful data).
 5. **Cross-correlation (X)**: Multiplies frequency data from each telescope pair.
 6. **Accumulation**: Connects samples to form final visibility products.
- **Dataflow Model**:

$$G = \langle A, F \rangle$$

where A is the set of actors (computations) and F is the set of FIFO buffers.

- Offers a structured approach to manage **parallelism** and hardware **deployment**.

Dataflow resource allocation for CPU-GPU systems

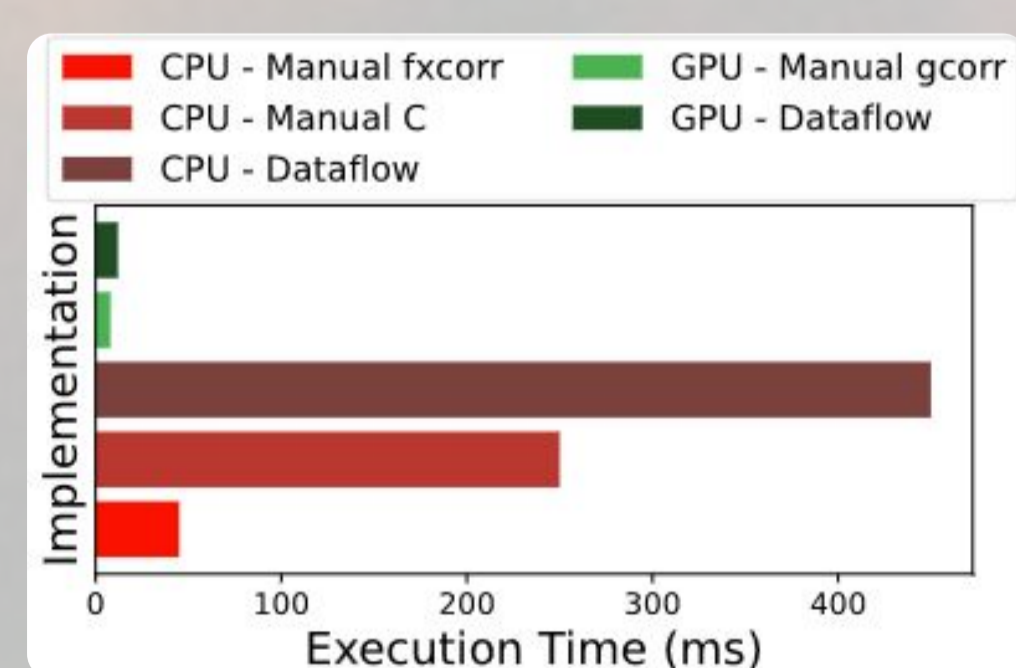


- **GPU-friendly** computation extraction based on **pattern matching**
- Mapping considering offloading gains:
$$\text{map} = \begin{cases} \text{CPU}, & \text{if } \text{Time}_{\text{CPU}}(\text{sub}) > \text{Time}_{\text{GPU}}(\text{sub}) \\ \text{GPU}, & \text{otherwise} \end{cases}$$
- Code translation based on target (CPU → C, NVIDIA GPU → CUDA)

Result

	SrDAG	Allocation
STD(1c) / M(1c,1g)	93.67	58088
STD(8c) / M(1c,1g)	93.67	230509
SCAPE(1c) / M(1c,1g)	2.55	13.83
SCAPE(8c) / M(1c,1g)	5.64	38.00


Average reduction factor: the existing methods (STD) over the proposed method (M).
 1c: 1 core, 1g: 1 GPU.
 The larger, the better.



Execution time by DiFX implementation. The lower, the better.

- ✓ **Significant** reduction by clustering, reflected in the acceleration of the resource allocation process.
- ✓ **Speedup** resource allocation by a factor of **13** compared to the standard methods.
- ✓ Optimized dataflow implementations **compete** with manual ones.

Summary conclusion

- Achievement:
 - ✓ **Automated dataflow resource allocation method and code generation** for heterogeneous CPU-GPU systems.
 - ✓ **Fast mapping** based on computation behavior.
 - ✓ Implemented into **PREESM**  and available on **Github**.
- Future work:
 - Scale it up on **multi-node** modern HPC systems.
 - Reorganize scheduling based on component cache (CPU-GPU) integrating **polyhedral optimization**.

