RADIOBLOCKS: Using GPUs for signal processing in the AI era

Mark Kettenis (JIVE)
for the RADIOBLOCKS WP4 team

ADASS XXXVI, November 5 2026, Boorloo/Perth (Australia)

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement number 101093934

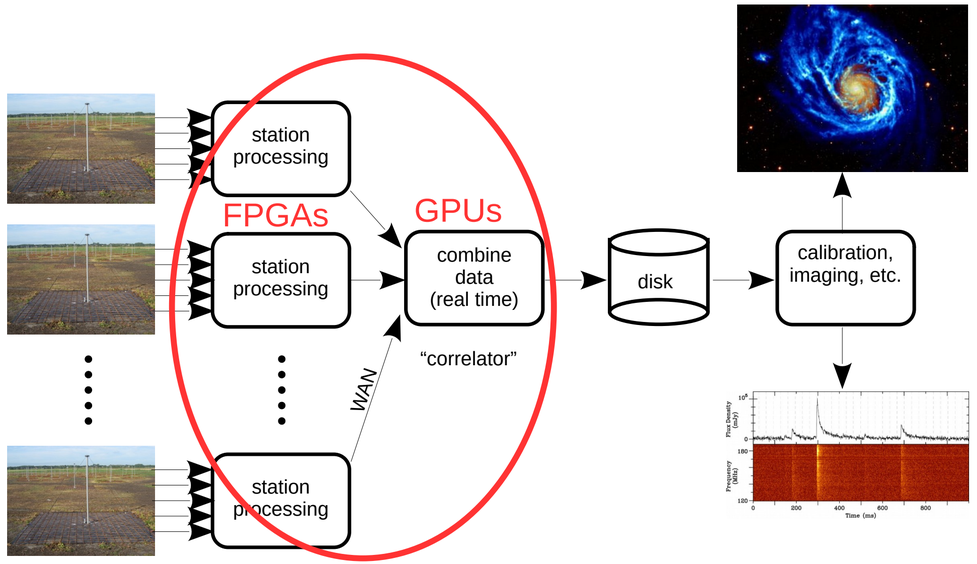
RADIOBLOCKS

- Horizon Europe funded project to develop technological components for the entire data chain
- Running from March ’23 —February ’27
- Work packages:
  - WP2 novel detectors and components
  - WP3 digital receivers
  - WP4 data transport and correlation
  - WP5 data processing toolkit for advanced radio astronomy
- Increase TRL of several new technologies

RADIOBLOCKS WP4 Goals

data transport and correlation

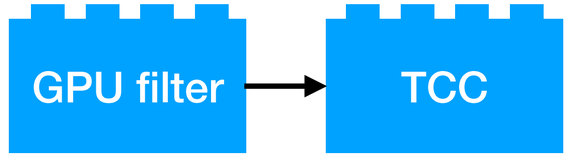
- Develop common “Radio Blocks”
  - Code re-use between applications
  - Reduce code maintenance burden
  - **Improve energy efficiency!**
- Develop methods for 400Gb/s FPGA -> GPU communication
- Integrate radio blocks & I/O methods in correlator / beam-former applications

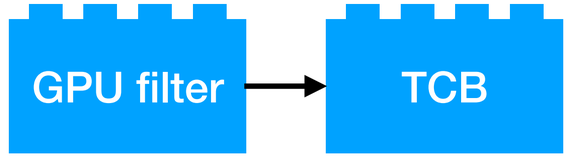


Radio Blocks

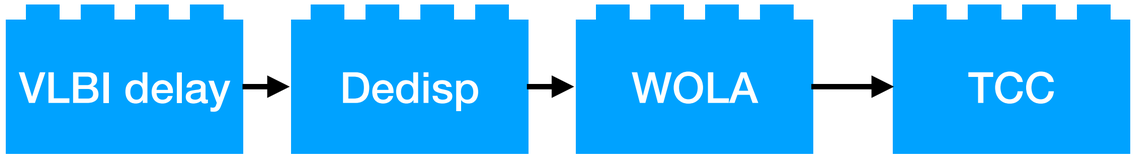
- Open Source GPU (mostly) processing libraries:

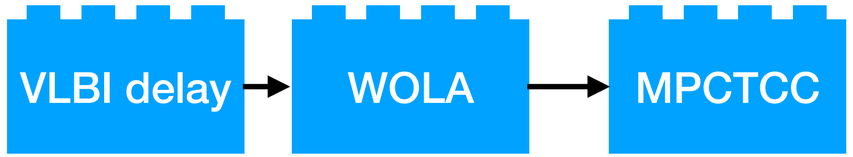
GPU filter	https://git.astron.nl/RD/gpu-filter
VLBI delay tracking + WOLA	https://code.jive.eu/keimpema/sfxgpu
Coherent de-dispersion filter	https://code.jive.eu/keimpema/sfxgpu
Tensor Core correlator	https://git.astron.nl/RD/tensor-core-correlator
Tensor Core beam former	https://git.astron.nl/RD/recruit/ccglib
MPC Tensor Core correlator	https://code.jive.eu/kettenis/mpc-tensor-core-correlator





- Blocks hide complexity; users can focus on the application
- Somewhat coherent APIs (facilitated by cudawrappers https://github.com/nlesc-recruit/cudawrappers)
- Easily integrated in Cmake-based builds
- Some of these libraries are used outside of radio astronomy!





RADIOBLOCKS cluster

- Development cluster with many different accelerators:
  - Grace-Hopper (GH200), DGX Spark (GB10), AGX Orin/Thor, RTX 6000 Blackwell, RTX 4000 Blackwell, W7700 (and hopefully R9700)
  - 800 Gbit/s Ethernet

Investigate impact of:

- Unified vs. non-unified memory
- Edge vs. Server
- HBM vs. GDDR memory
- arm64 vs. amd64



Technology developments

Lower-precision data types

- Traditional data types:
  - Integer data types: INT8, INT32, INT64
  - Floating-point data types: FP32, FP64
- New data types:
  - Integer data types: INT4
  - Floating-point data types: FP16, BF16, FP8 (E4M3 or E5M2), FP4
- Benefits:
  - Lower memory bandwidth
  - Lower memory usage
  - Higher performance (limited)

Good enough to represent noisy signals?

VLBI typically uses 2-bit samples!

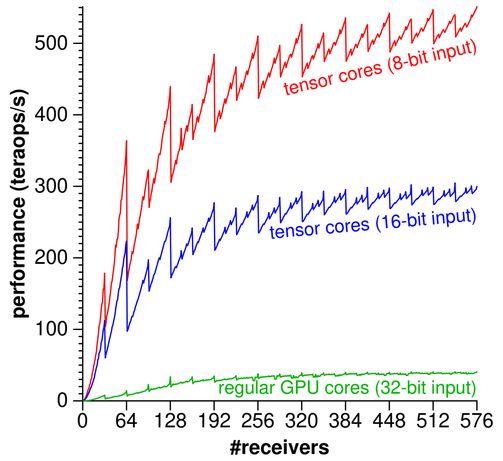
But think carefully about scaling!

Technology developments

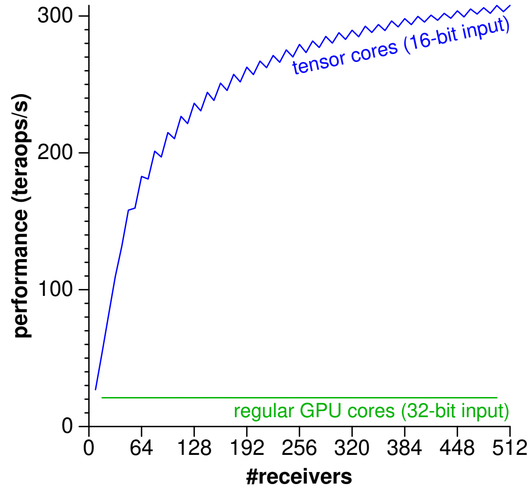
Tensor Cores (and equivalent technology)

- Dedicated cores for Matrix Multiply Accumulate operations
  - Use lower-precision types as input
  - Accumulate in higher range/precision types
- Extremely power efficient
- Used in Tensor Core correlator and Tensor Core beam former radio blocks

Tensor Core correlator



Tensor Core beamformer



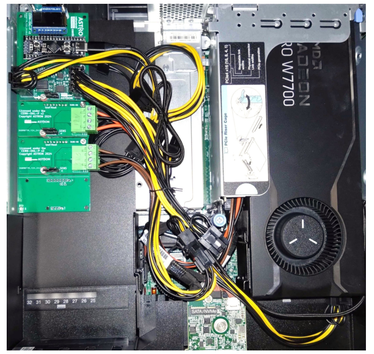
Technology developments

cuFFTdx

- cuFFT implements GPU kernel for each transform
  - Requires input and output in GPU memory
- cuFFTdx works inside a GPU kernel
  - Can be combined with other operations in a single kernel
  - Device specific limits on number of points and parameters (typically 32k for power-of-two, 64k on high-end GPUs)
- Used by almost all other radio blocks!

Power Measurement

- Power Measurement Toolkit (PMT):
  - Uniform interface for power-related performance counters (CPU, GPU, SoC)
  - Not always high-enough resolution for optimizing applications
- PowerSensor 3.0:
  - Hardware developed by ASTRON
  - Directly monitors PCIe power supply
  - Supported in the PMT



(a) PowerSensor3 attached to an AMD W7700 GPU.

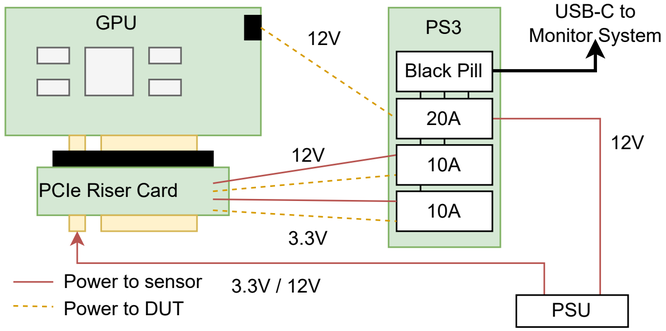
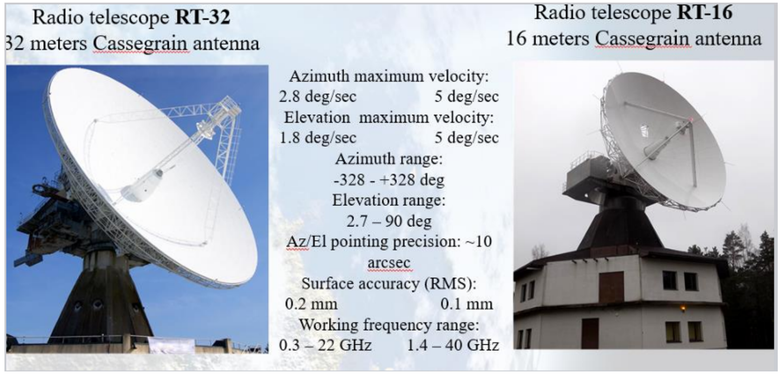


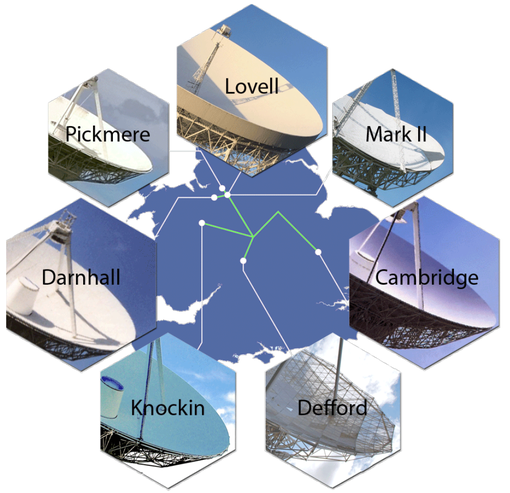
Fig. 1: Schematic of a PowerSensor3 measurement setup.

Applications

- COBALT correlator and beam former for LOFAR 2.0
- AARTFAAC correlator for LOFAR 2.0
- ISBI-AARTFAAC correlator for Irbene Single baseline Interferometer
- SFXGPU VLBI correlator
- TPGS: ALMA Total Power GPU Spectrometer
- New correlator for e-MERLIN (under investigation)

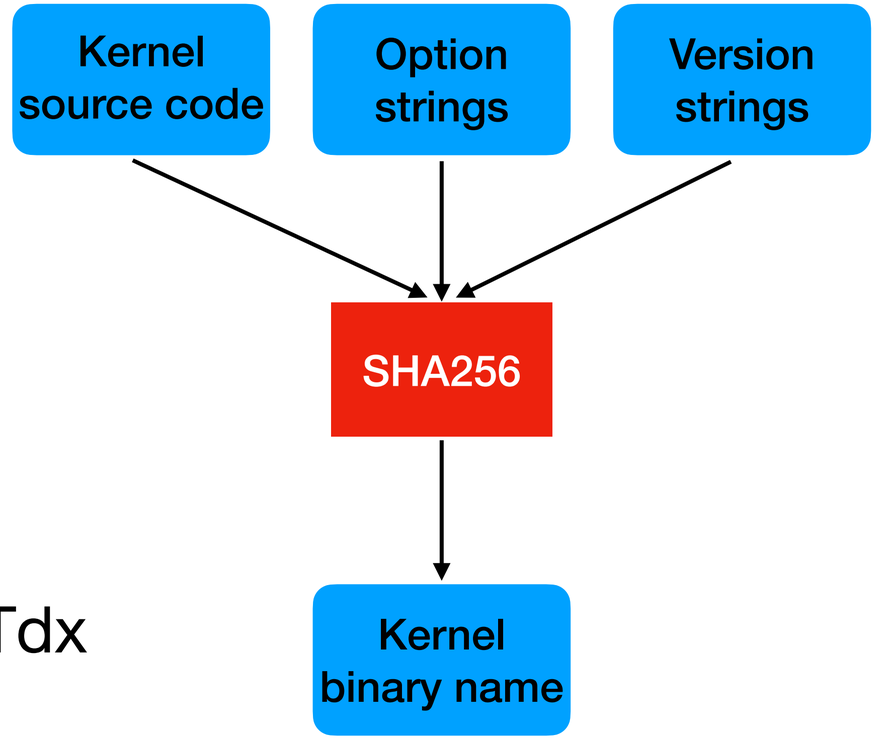
And other applications outside RADIOBLOCKS
(e.g. Jay Smallwood's talk on Monday)





Optimization techniques

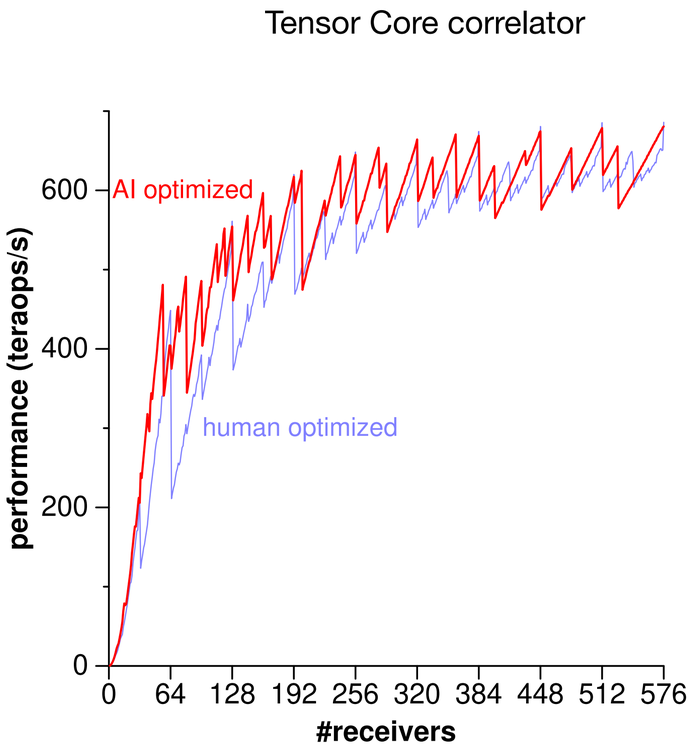
- Fix parameters at compile time
  - Frees up GPU registers for actual compute
- Use dynamic compilation
  - To support varying those fixed parameters
- Use a kernel cache
  - To avoid expensive recompilation
    - Really expensive for heavily templated cuFFTdx



AI assisted coding

Experimentation

- Optimization of some radio blocks
- Support for AMD GPUs:
  - Added HIP support in some radio blocks
  - Improved HIP support in cudawrappers to support AMD and NVIDIA at the same time
- AARTFAAC correlator distributing load over AMD and NVIDIA GPUs within single application
- Many new branches that need to be integrated…
  - How do we do this without creating unmaintainable black boxes?





This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement number 101093934