S5302 - OPTIMIZATION OF GPU-BASED SIGNAL PROCESSING OF RADIO TELESCOPES

VINAY DESHPANDE
DEVELOPER TECHNOLOGY
NVIDIA

HARSHAVARDHAN REDDY
ENGINNER, NCRA

INTRODUCTION

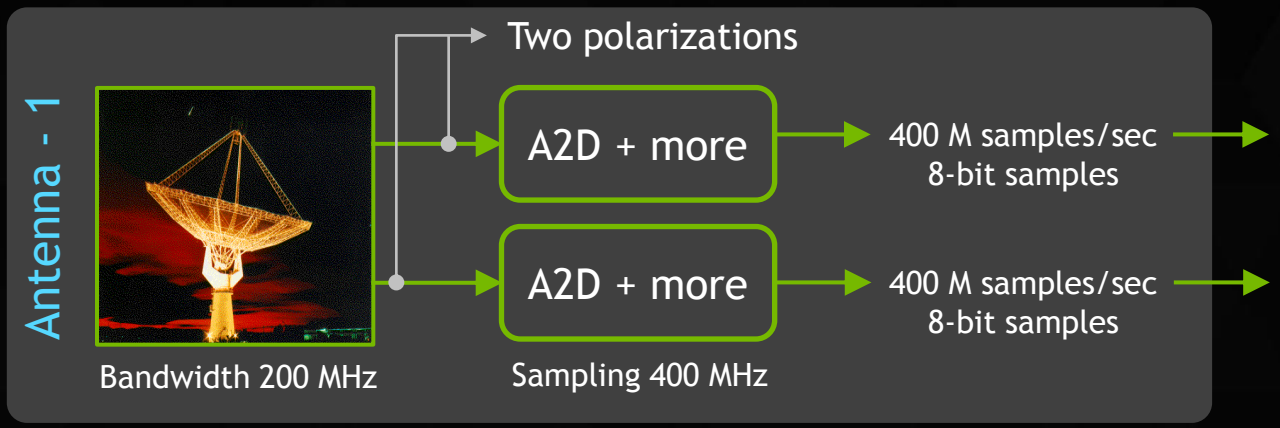
- NCRA – National Center for Radio Astrophysics
  - Pune, India.
  - http://ncra.tifr.res.in/ncra
- GMRT – Giant Meterwave Radio Telescope
  - Situated at Kodad near Pune, India.
  - http://gmrt.ncra.tifr.res.in/
  - Consists of 30 dish antennas
  - 45 m diameter each, spread over 25 Km
  - Used by radio-astronomers world-wide

uGMRT EFFORT

- The GMRT backend has been upgraded recently
  - The “uGMRT”
- Key change: Bandwidth 32 -> 200/400 MHz
  - Prototype system with 16 antennas – 8 compute nodes up and running
  - GPUs upgrade from Fermi to Kepler
- Optimizing software backend
  - For better science, less power and reduction in cost
  - On going work involving NVIDIA and NCRA teams
- Contribution towards SKA

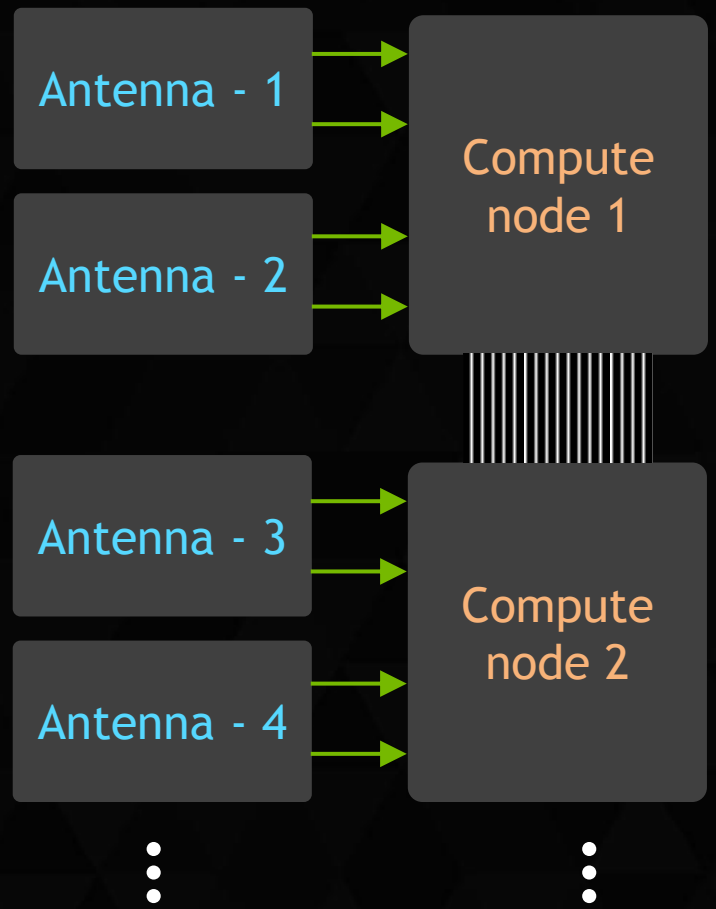
GMRT BACKEND

- Each antenna has two polarizations
- If the antenna is operating at 200 MHz bandwidth
  - Sampling needs to be frequency 400 MHz
  - Produces 400 million samples/sec
  - 800 million samples per antenna per sec
  - Total 800 * 32 = 25.6 G samples/sec (2 additional signal sources for debug and test)
- Signal processing backend needs to process all these samples in real-time



BACKEND: COMPUTE INFRASTRUCTURE

- Samples from two antennas is fed to a single compute node
  - The number could change for other telescopes
  - Can be decided by I/O requirements
- 16 compute nodes
  - Connected over high-speed network
- Each compute node has
  - One CPU
  - One or two GPUs



GPU CORRELATOR

- Operations involved
  - Data format conversion (Unpacking)
  - Discrete Fourier Transform (DFT)
  - Phase Rotation
  - Multiply-And-Accumulate (MAC)

1. UNPACKING

- For converting each sample
  - 8-bit read (integer) and 32-bit write (floating point)
- Dominated by I/O
- Unpacking is immediately followed by DFT
  - 32-bit data per sample needs to be read again
- This read after write trip can be saved
  - cuFFT callbacks introduced in CUDA 6.5
- cuFFT callbacks can be used to combine unpacking with FFT operation
- Result - overhead of unpacking is reduced by 25%

2. DISCRETE FOURIER TRANSFORM

- DFT is implemented using cuFFT library APIs
- cuFFT Mode selection
  - R2C
  - C2C – Requires additional 2x2 Butterfly kernel
- Several possible combinations of input and output callback
  - Unpacking, Phase Rotation, 2x2 butterfly

	No callbacks	Unpacking callback	Phase Rotation	2x2 Butterfly callback
R2C	Tested	Tested, second best	Tested	NA
C2C	Tested	Tested, best	NA	Tested

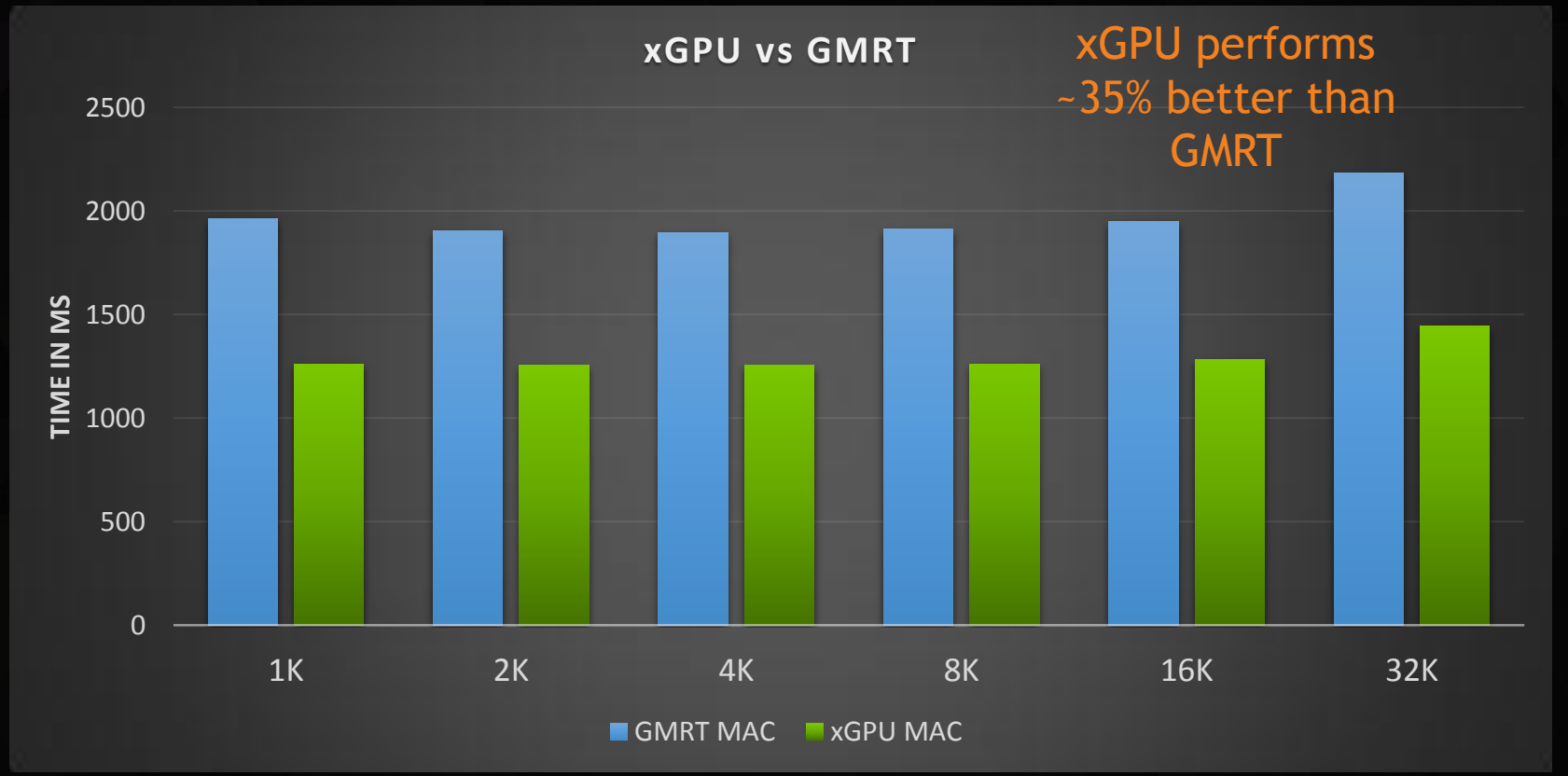
3. PHASE ROTATION

- Essentially multiplication by a constant
  - Constant depends on antenna, frequency channel and time slice
- The kernel computes each constant on-the-fly
  - Lots of math operations
- Redundancies in computation identified and removed
    - Improvement in performance 10%
- Switching from CUDA 6.0 to 6.5 boosted performance by 50%

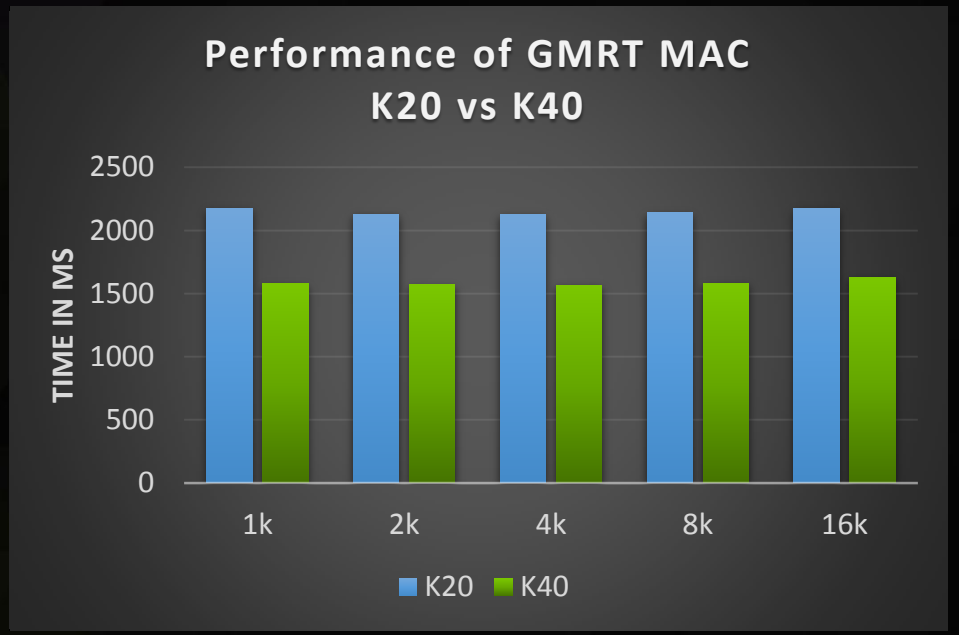
4. MAC

- The most costly operation
  - Cost grows proportional to (antenna)^2
- Choices for MAC routines
  - GMRT – original routine
  - xGPU – Mike Clark's highly optimized MAC library
- xGPU performs better is almost all cases
  - More so for higher number of antennas
- Side effect – Input/output reordering is required
  - (antenna, time, frequency) -> (time, frequency, antenna)
  - Shared memory based implementation achieves bandwidth of 128 GB/s on K20

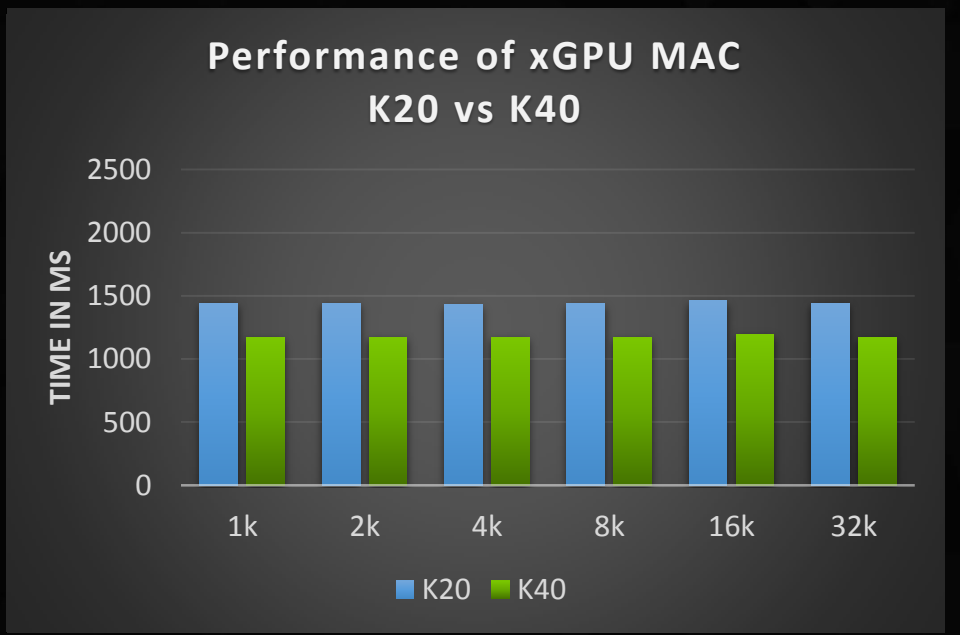
PERFORMANCE OF MAC



MAC KERNELS ON K40



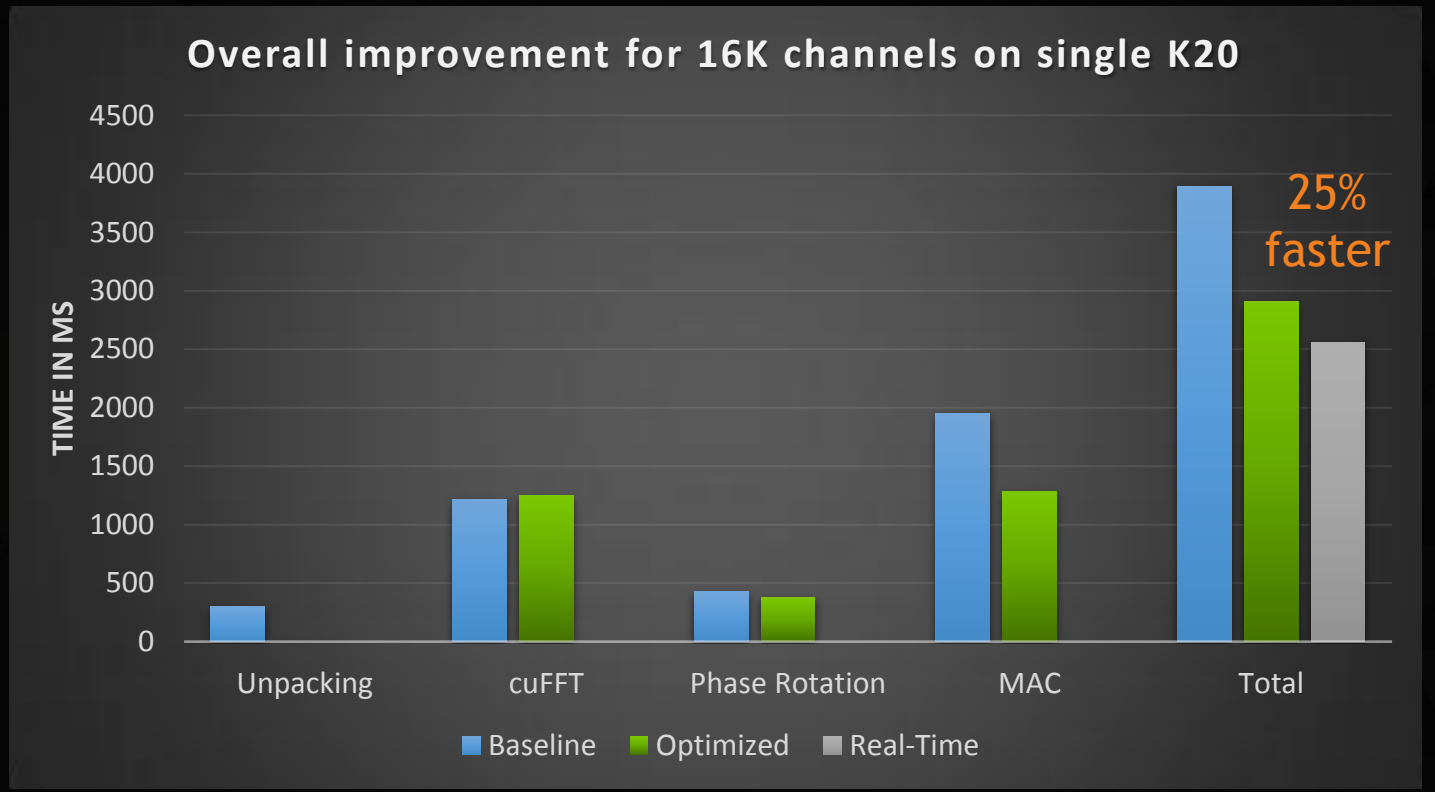
25-27% improvements



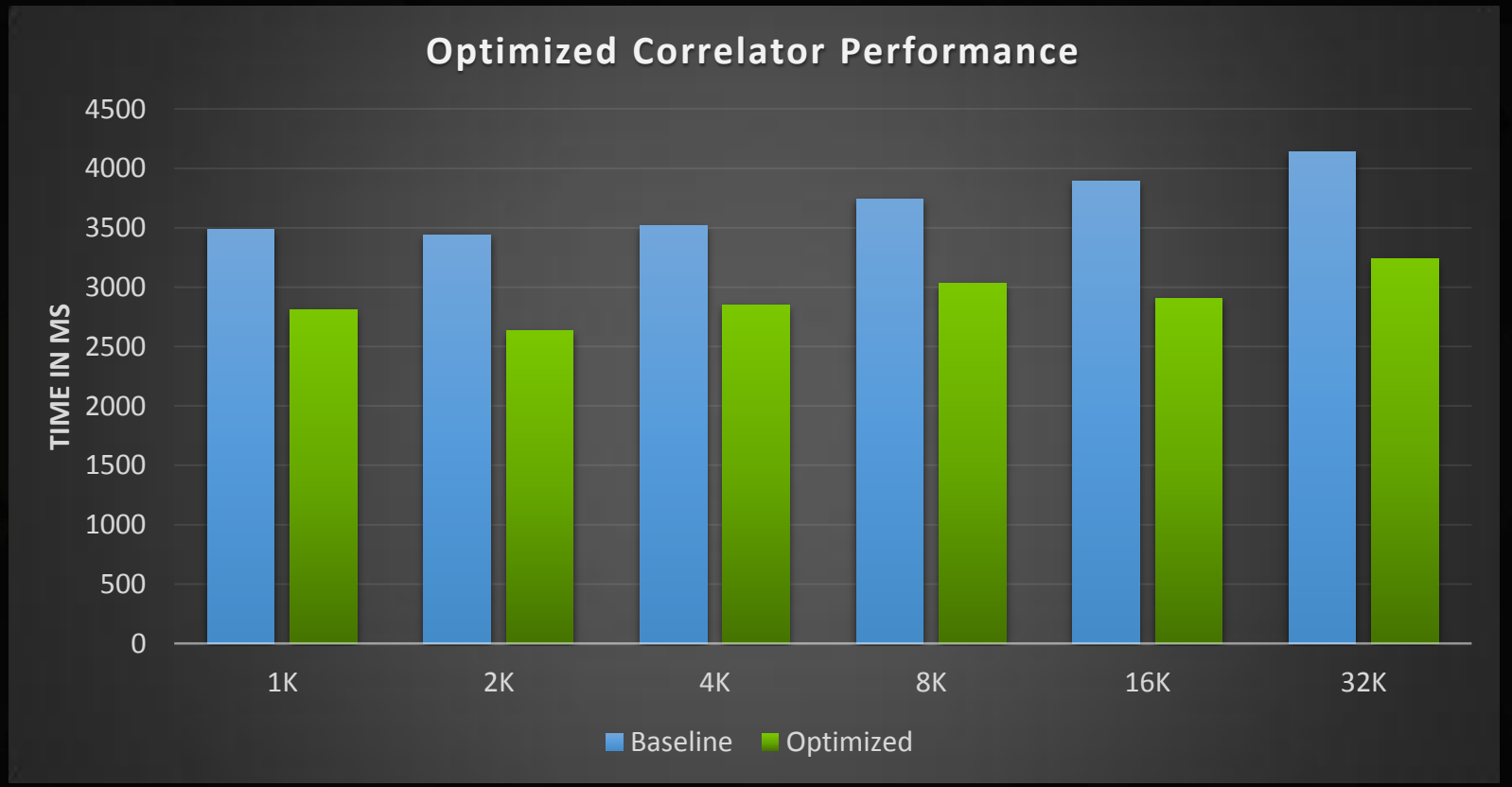
~18% improvements

OVERALL RESULTS

OVERALL IMPROVEMENTS



OVERALL IMPROVEMENTS

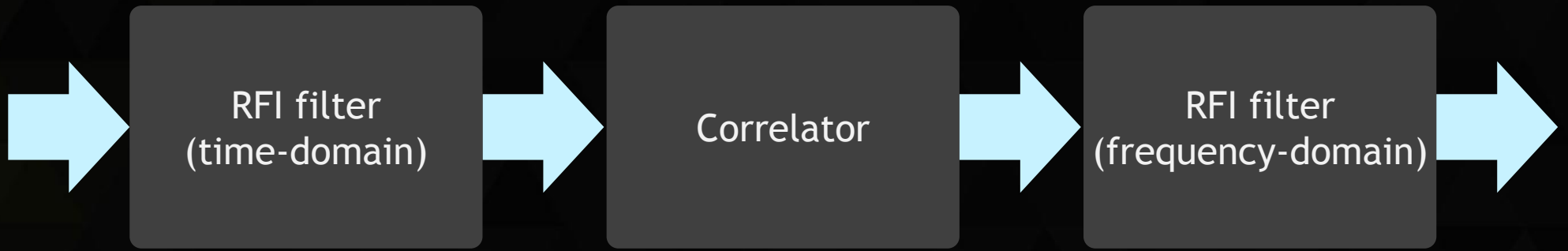


20-25% better performance

RFI REJECTION

RFI REJECTION

- RFI – Radio Frequency Interference
- RFI needs to be removed in real-time
- GMRT backend has time-domain RFI filtering implemented
  - Desirable to have RFI filtering in both domains



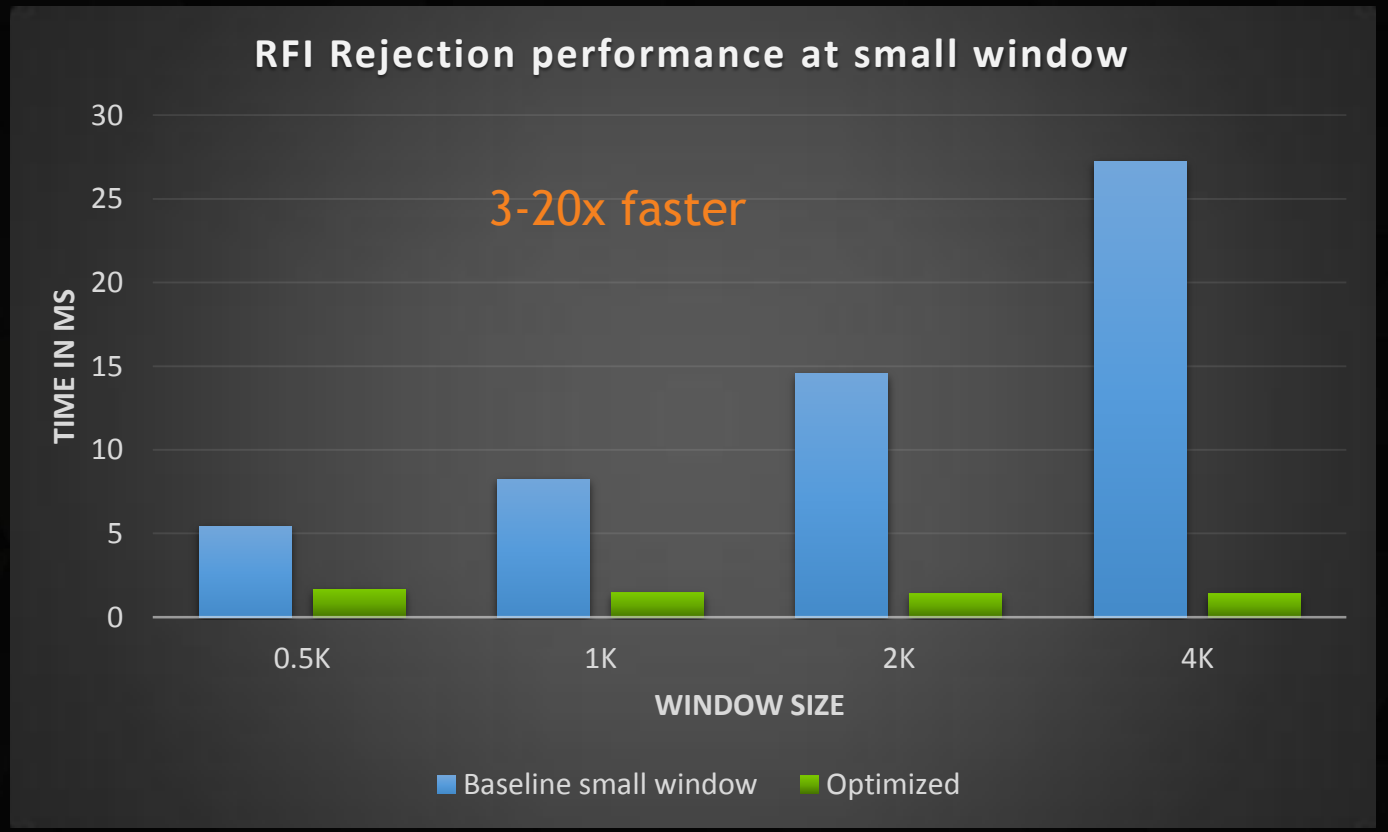
RFI REJECTION CODE

- GMRT implements Median Absolute Deviation (MAD) based filtering
  - MAD is a robust estimator
- Stream of input data is divided in fixed width windows
- For each window
  - First MAD is computed
  - Then threshold filter is applied
- All the windows can be processed concurrently
- GMRT has two implementations of the algorithm
  - Optimized for small window – (< 1K)
  - Optimized for large window – (> 4k)

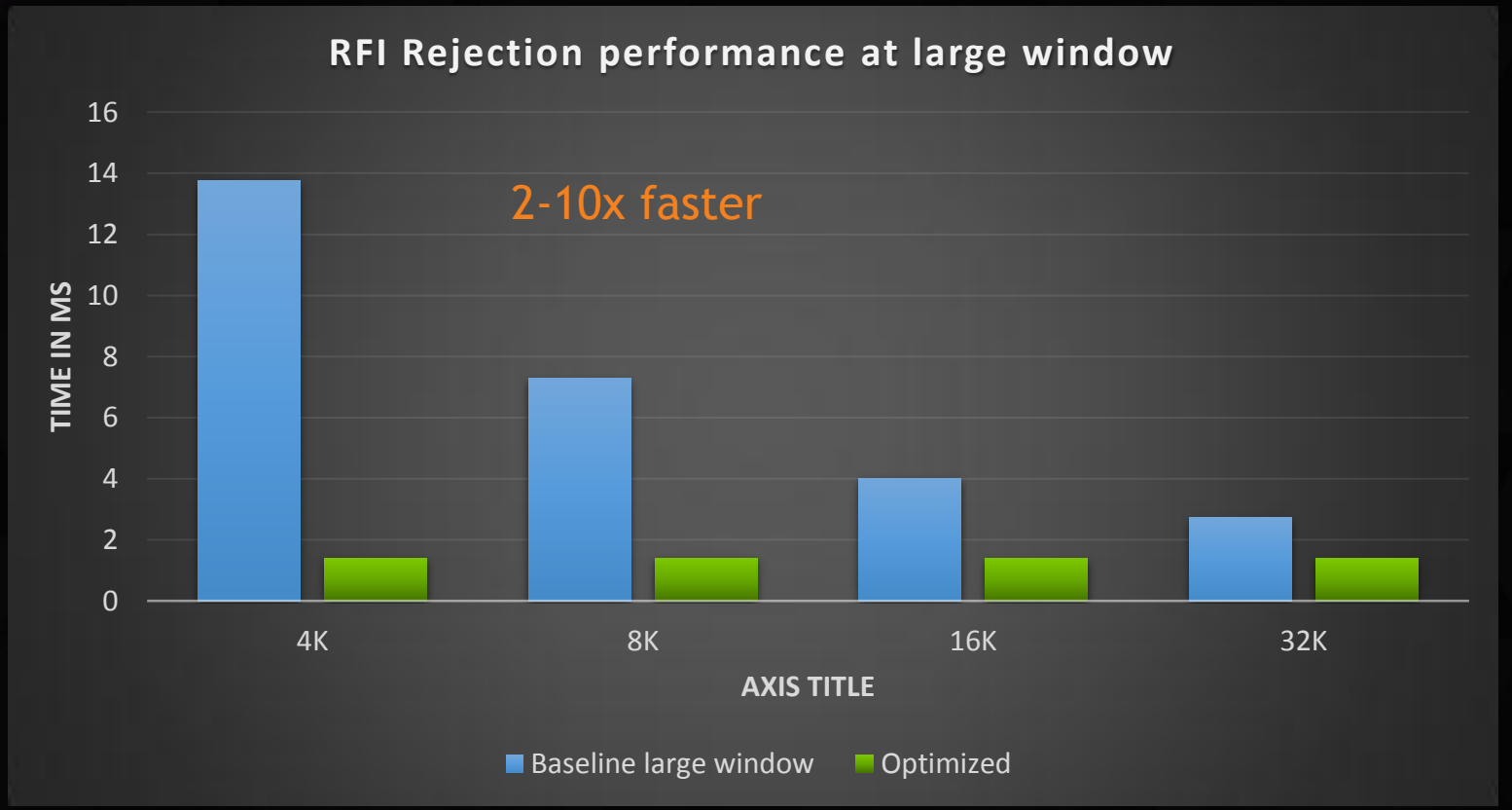
IMPROVEMENTS IN RFI FILTERING

- Implicit histogram computation
  - Second histogram is computed from first instead of re-fetching samples
- Integers instead of floating point numbers
  - $MAD = MAD_1 + \frac{MAD_2}{2}$
  - Helps in removing calls to ceil, floor etc.
- Reduced branching
  - 8 if-else blocks reduced to 4
- Reduction in launch latency overhead
  - Launching smaller number of bigger kernels
  - Side effect of combining kernels – temporary storage avoided
- Single version for all window sizes

RFI FILTERING RESULTS



RFI FILTERING RESULTS



REFERENCES

- S3225 - Powering Real-time Radio Astronomy Signal Processing with GPUs
  - GTC - 2013, Harshavardhan Reddy, Pradeep Gupta
- S4538 - Real-Time RFI Rejection Techniques for the GMRT Using GPUs
  - GTC 2014, Rohini Joshi
- NCRA-NVIDIA collaboration work report phase 1 and phase 2

ACKNOWLEDGEMENT

- Team NCRA
  - Dr. Yashwant Gupta
  - Harshavardhan Reddy
  - Rohini Joshi
  - Niruj

THANK YOU

JOIN THE CONVERSATION

#GTC15