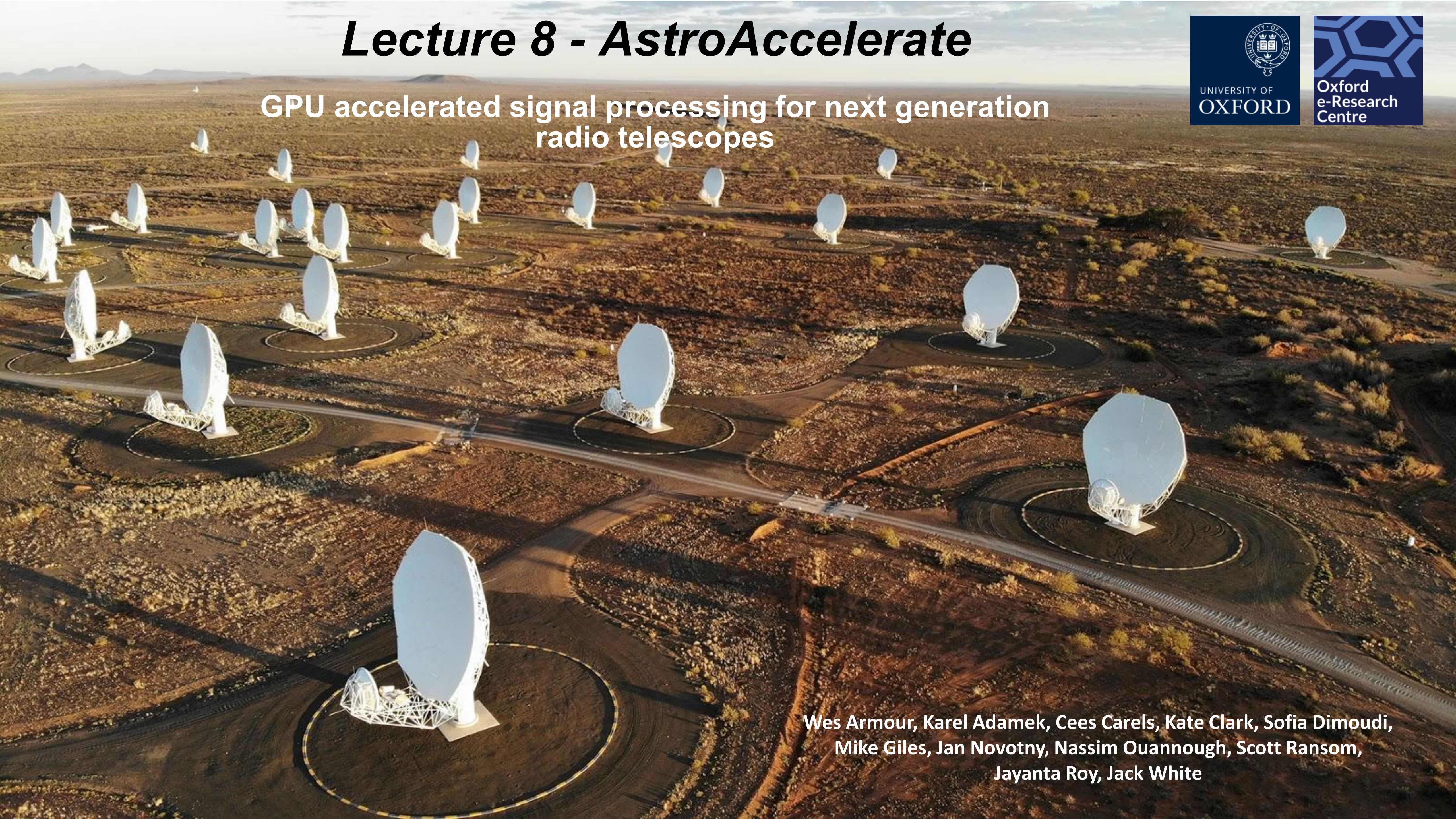


Lecture 8 - AstroAccelerate

GPU accelerated signal processing for next generation radio telescopes

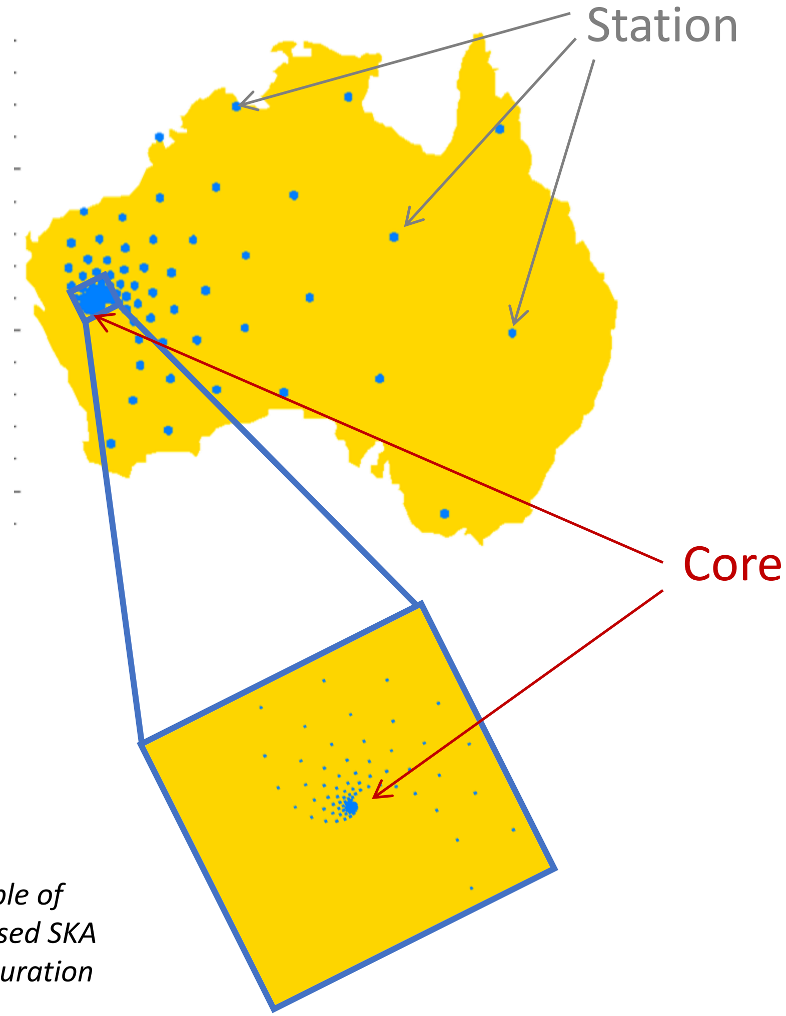
Wes Armour, Karel Adamek, Cees Carels, Kate Clark, Sofia Dimoudi, Mike Giles, Jan Novotny, Nassim Ouannough, Scott Ransom, Jayanta Roy, Jack White

Part One

SKA
SQUARE KILOMETRE ARRAY

A brief introduction to

What is SKA?



Example of proposed SKA configuration

What is SKA?

*SKA is a **ground based radio telescope** that will span continents.*

What does SKA stand for?

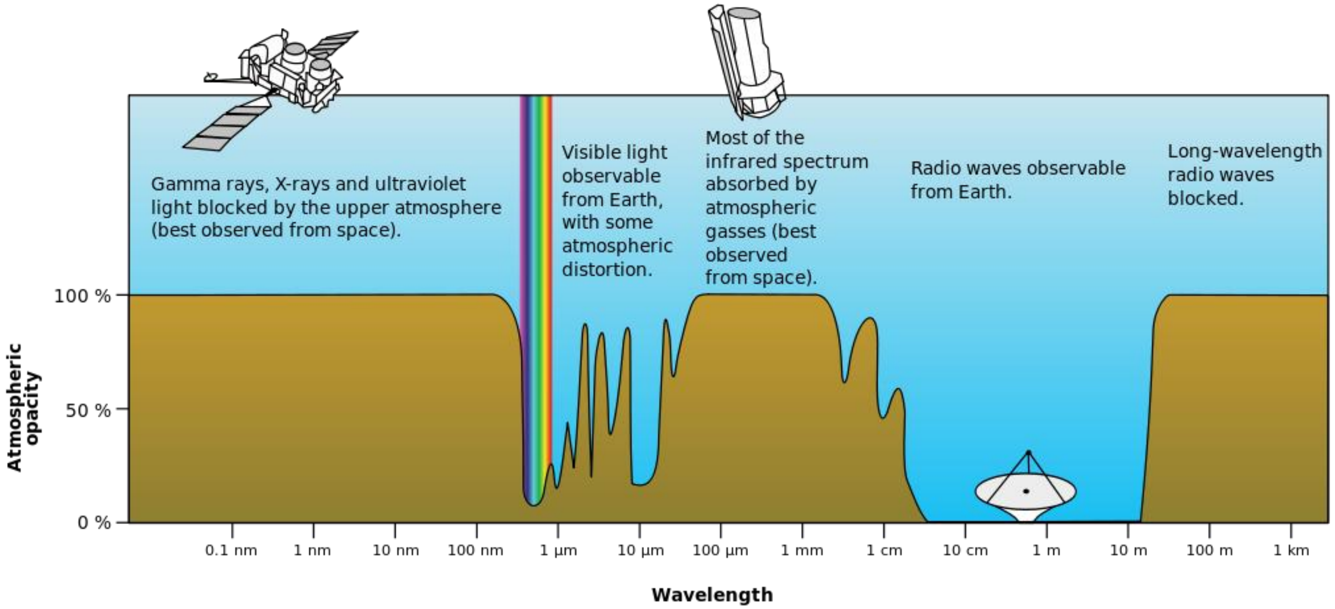
*Square Kilometre Array, so called because it will have an **effective collecting area of a square kilometre**.*

Where will SKA be located?

*SKA will be built in **South Africa and Australia.***

Graphic courtesy of Anne Trefethen

What is SKA?



SKA is a ground based telescope. This means that it is most sensitive to the radio range of frequencies. The radio range of frequencies that can be observed from here on Earth is very wide, specifically SKA will be sensitive to frequencies in the range of 50MHz to 20GHz (**wavelengths 15 mm to 6 m**). This makes SKA ideal for **studying lots of different science cases.**

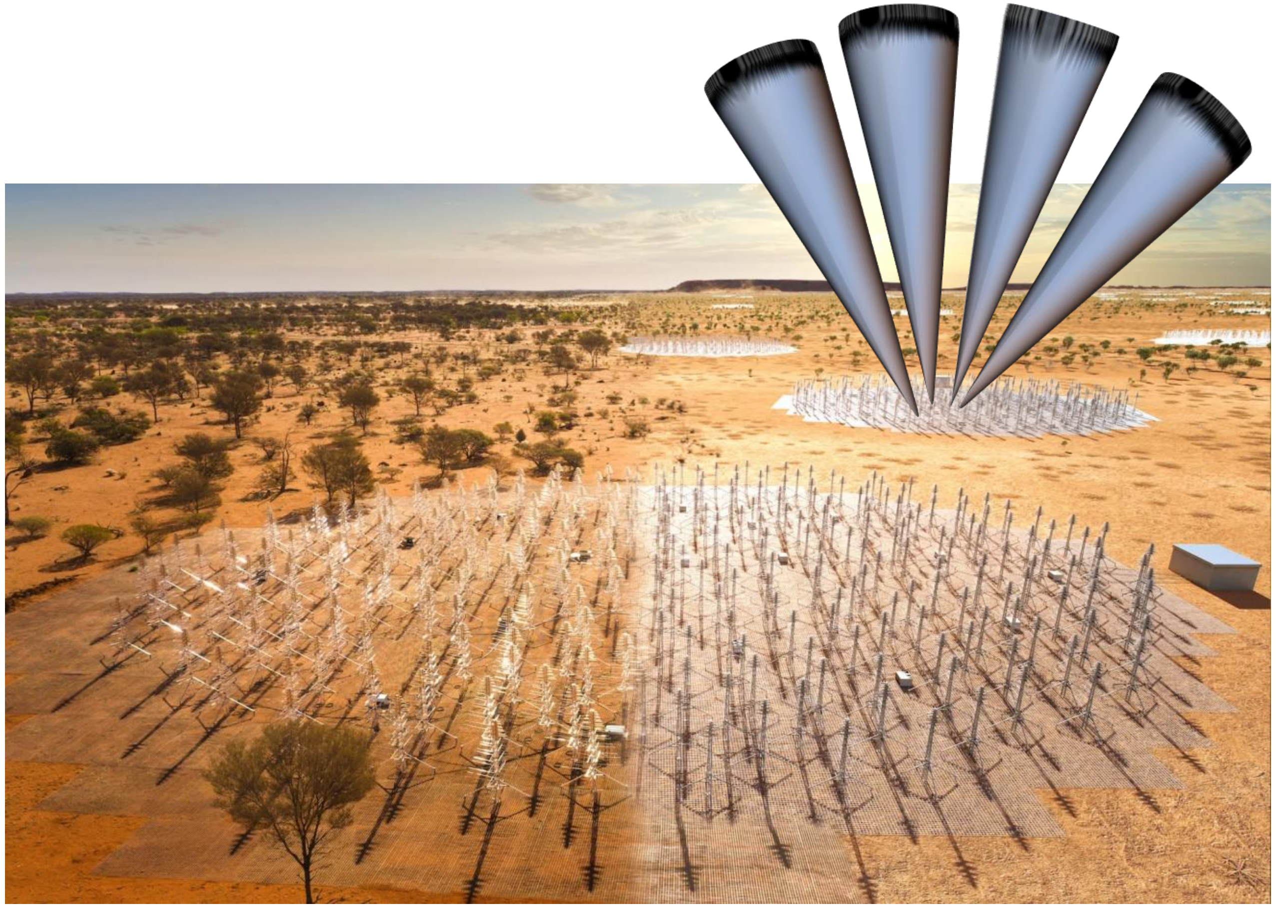
Image source Wikipedia. Authors: NASA (original); SVG by Mysid

What is SKA?

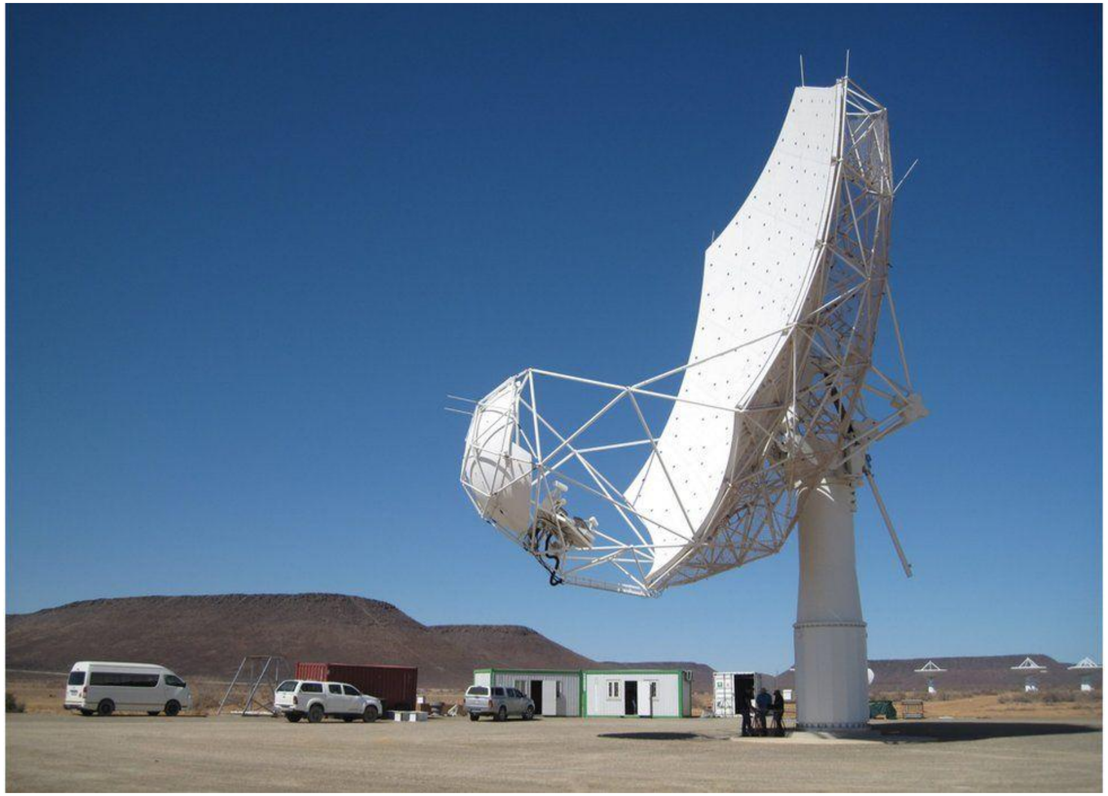
SKA will have the ability to use all of its antennas to **produce images of the radio sky in unprecedented accuracy and detail.**

It will also be able to use combinations of antennas to perform **multiple observations of different regions of the sky at the same time**.

In this scenario data from each beam can be computed in parallel.



SKA Science



SKA will study a wide range of science cases and aims to answer some of the fundamental questions mankind has about the universe we live in.

- How do galaxies evolve
  - – What is dark energy?
- Tests of General Relativity
  - – Was Einstein correct?
- Probing the cosmic dawn
  - – How did stars form?
- The cradle of life
  - – Are we alone in the Universe?

SKA time domain - signal processing

Time Domain Team

The time domain team is an international team led by Oxford and Manchester.

It aims to deliver an end-to-end signal processing pipeline for time domain science performed by SKA (see right).

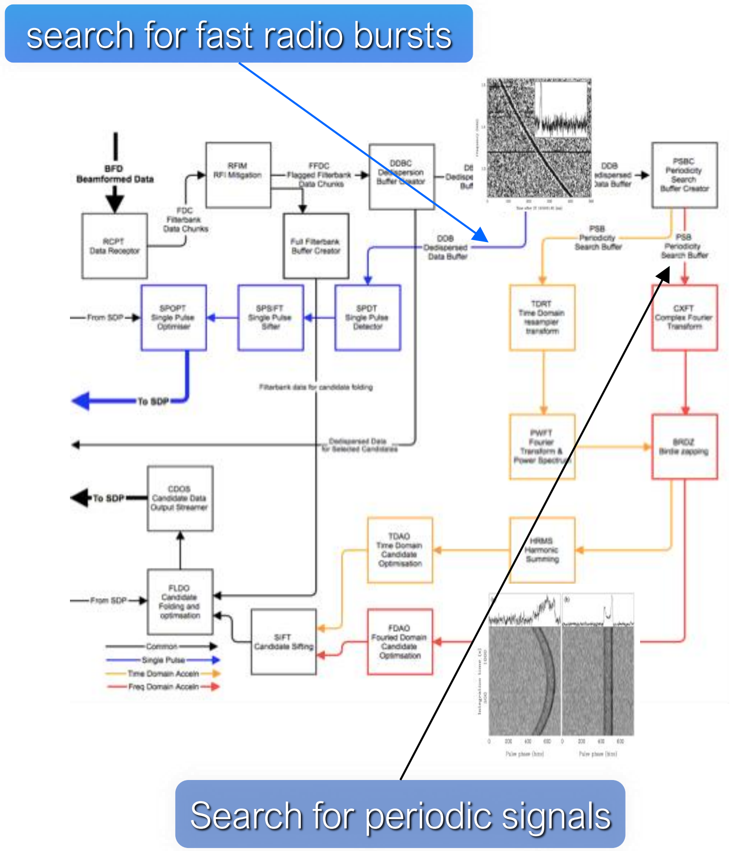


Image courtesy of Aris Karastergiou

SKA time domain - signal processing

Our work focused on vertical prototyping activities.

We delivered accelerated algorithms for many-core technologies, such as GPUs to perform the processing steps within the signal processing pipeline with the aim of achieving real-time processing for the SKA.

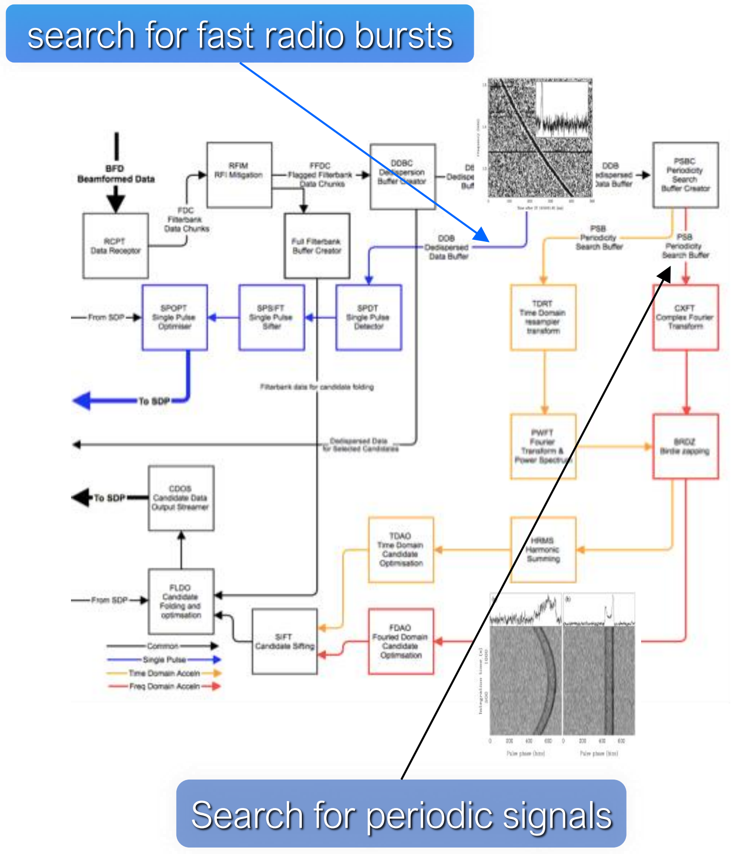


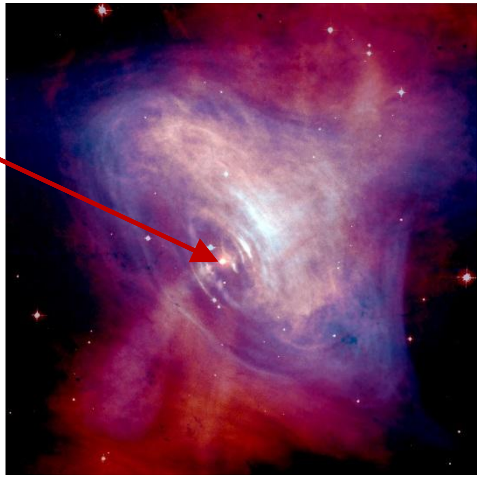
Image courtesy of Aris Karastergiou

SKA time domain science - Pulsars

Pulsars are magnetized, rotating neutron stars. They emit synchrotron radiation from the poles, e.g. Crab Nebula.

Their magnetic field is offset from the axis of rotation as such (as observed from here on Earth, they act as cosmic lighthouses.

They are extremely periodic and so make excellent clocks!



Hester et al.

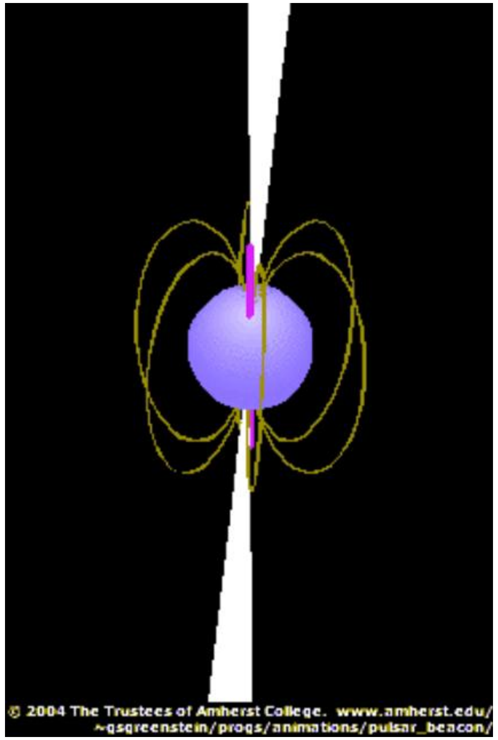
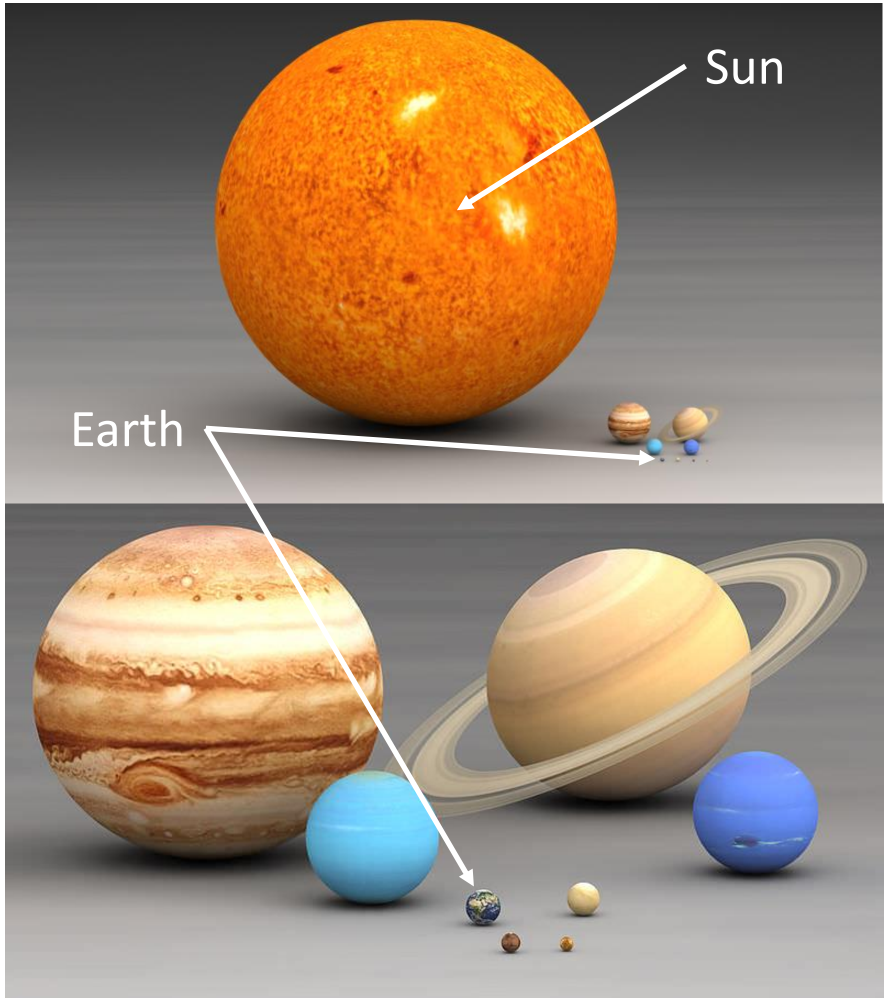


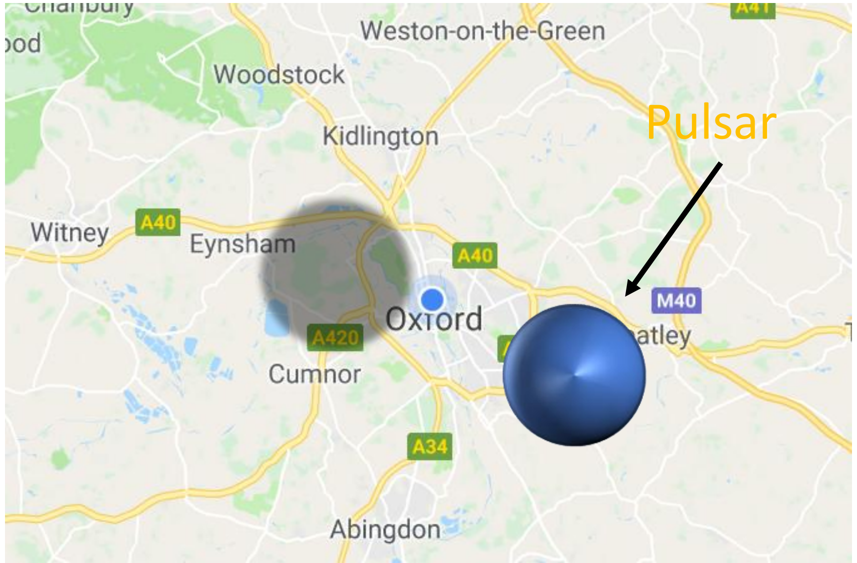
Image: Amherst College

Pulsars – size and scale



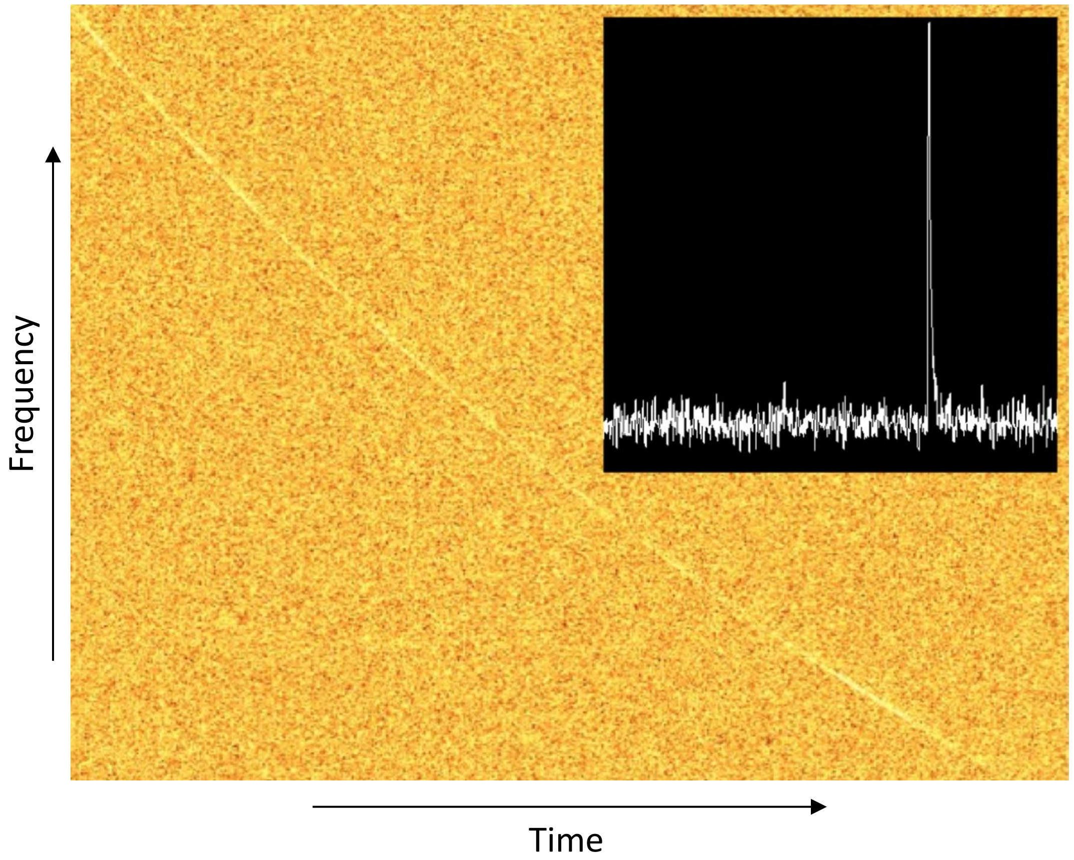
Pulsars are typically 1-3 Solar masses in size, they have a diameter of 10-20 Kilometres and a pulse period ranging from milliseconds to seconds.

Meaning that they are very small, very dense and rotate extremely quickly.



https://commons.wikimedia.org/wiki/File:Planets_and_sun_size_comparison.jpg (Author: Lsmpascal)

SKA time domain science - FRBs



Fast Radio Bursts (FRBs), were first discovered in 2005 by Lorimer et al.

They are observed as **extremely bright single pulses that are extremely dispersed** (meaning that they are likely to be far away, maybe extra galactic).

Now hundreds of FRBs have been observed or found in survey data. They are of unknown origin, but **likely to represent some of the most extreme physics in our Universe.**

Hence they are extremely interesting objects to study.

Credit: FRB110220 Dan Thornton (Manchester)

SKA time domain – data rates

The SKA will produce vast amounts of data. In the case of time-domain science we expect the telescope to be able to place ~2000 observing beams on the sky at any one time (there are trivially parallel to compute).

The telescope will take 20,000 samples per second for each of those beams and then it will measure power in 4096 frequency channels for each time sample. Each of those individual samples will comprise of 4x8 bits, although we are only really interested in one of the 8 bits of information.

Doing the math tells us that we will need to process 160GB/s of relevant data. This is approximately equal to analysing 50 hours of HD television data per second.

The most costly computational operations in data processing pipeline are

DDTR ~ $O(n_{dms} * n_{beams} * n_{samps} * n_{chans})$

FDAS ~ $O(n_{dms} * n_{beams} * n_{samps} * n_{acc} * \log(n_{samps}) * 1/t_{obs})$

Requiring ~2 PetaFLOP of Compute!

SKA time domain – data challenges

Because we would like to **monitor interesting and exotic events as they occur, we need to process data in real-time** (or as near to as possible).

So, storing the data and processing later isn't feasible. The data rates mean transporting data offsite would be challenging and costly.

Hence processing must happen close to the telescope. But how do we put a computer capable of processing big-data streams in real-time close to the telescope?

Connectivity, power, operation all pose significant problems.

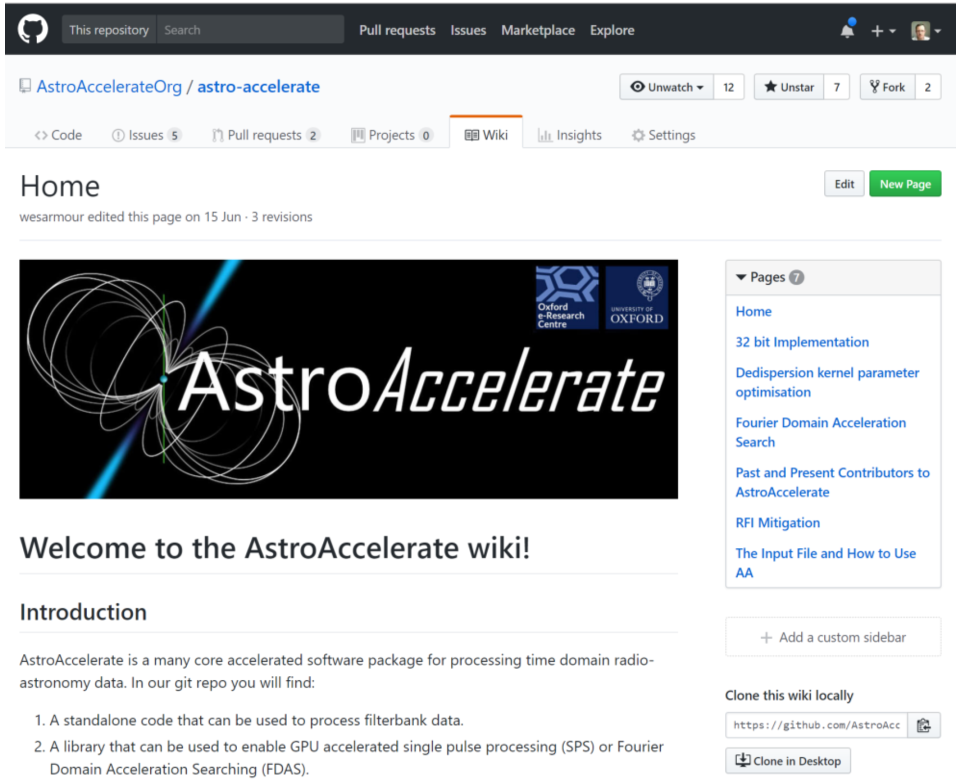
Part Two

Astro*Accelerate* – A case study

AstroAccelerate

AstroAccelerate is a GPU enabled software package that focuses on achieving real-time processing of time-domain radio-astronomy data. It uses the CUDA programming language for NVIDIA GPUs.

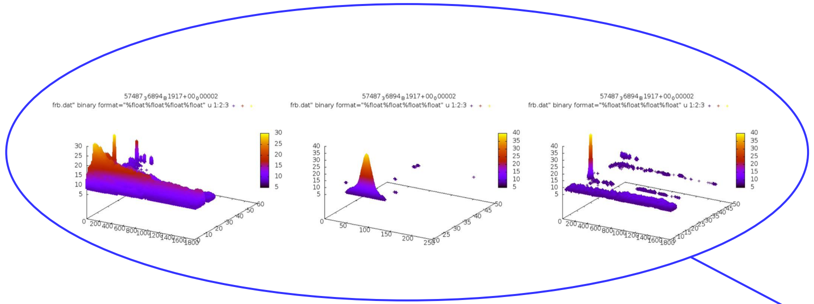
The massive computational power of modern-day GPUs allows the code to perform algorithms such as de-dispersion, single pulse searching and Fourier Domain Acceleration Searching in real-time on very large data-sets which are comparable to those which will be produced by next generation radio-telescopes such as the SKA.



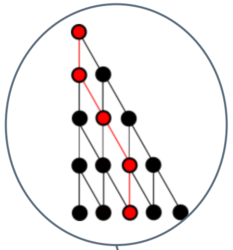
https://github.com/AstroAccelerateOrg/astro-accelerate

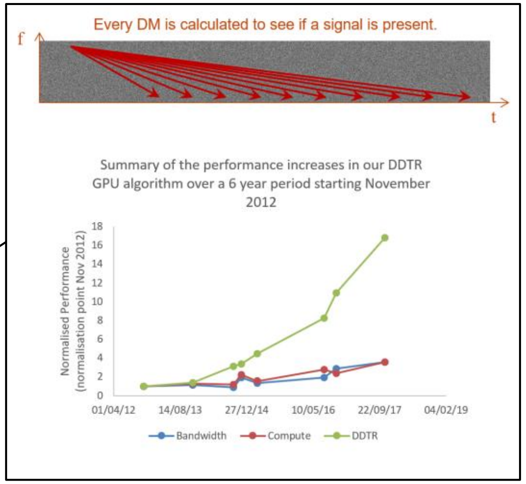
AstroAccelerate - Signal Processing

Radio Frequency Interference Mitigation



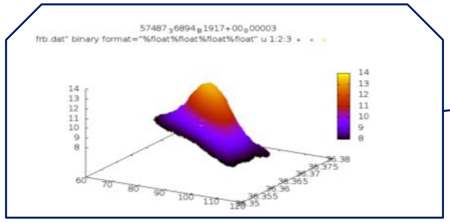
Harmonic Sum

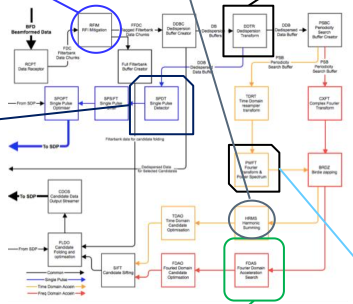




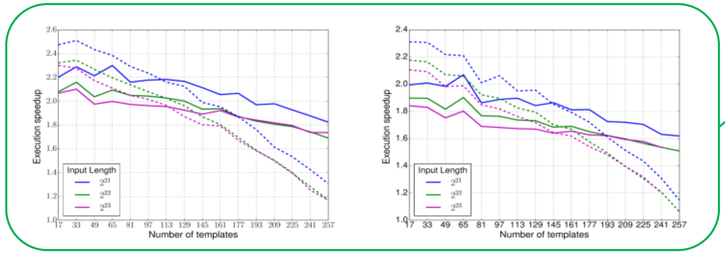
De-dispersion

Single Pulse Search

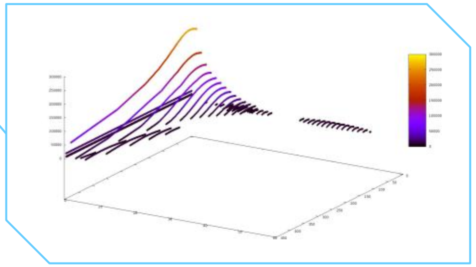




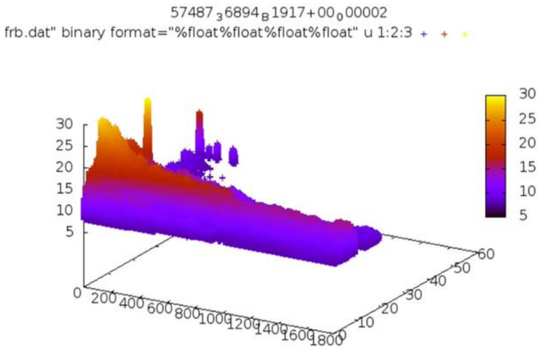
Fourier Domain Acceleration search

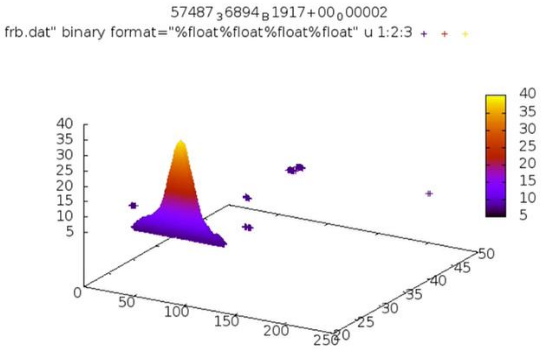


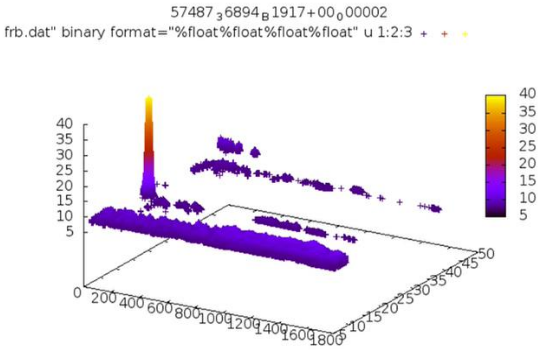
Periodicity Search



AstroAccelerate - Features

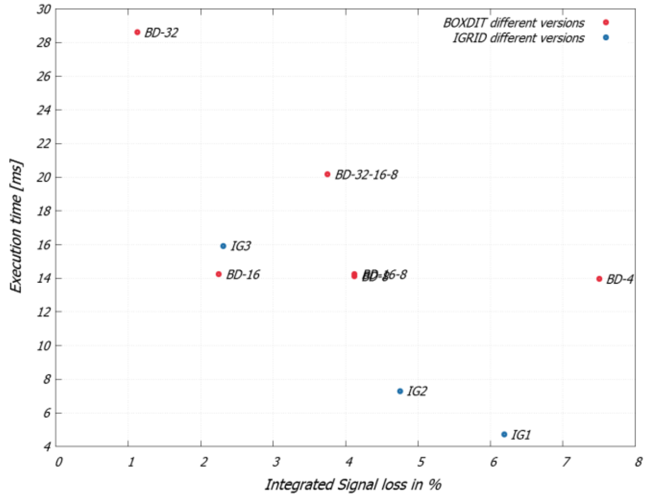






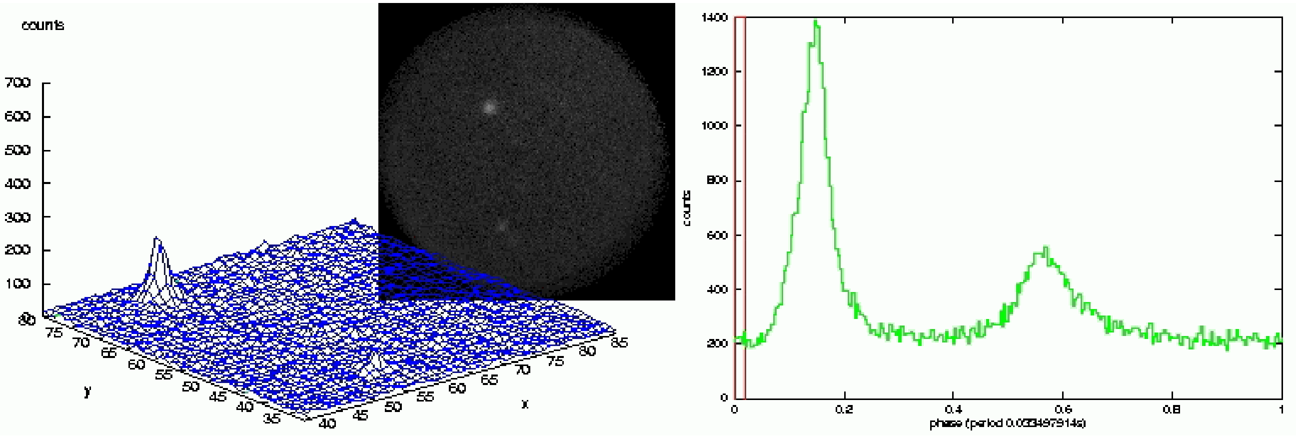
AstroAccelerate has the following features…

- Zero DM and basic RFI Mitigation
- DDTR
- Single Pulse Search
- Fourier Domain Acceleration Search
- Periodicity search with harmonic sum



Case study: Fourier Domain Acceleration Searching for the SKA

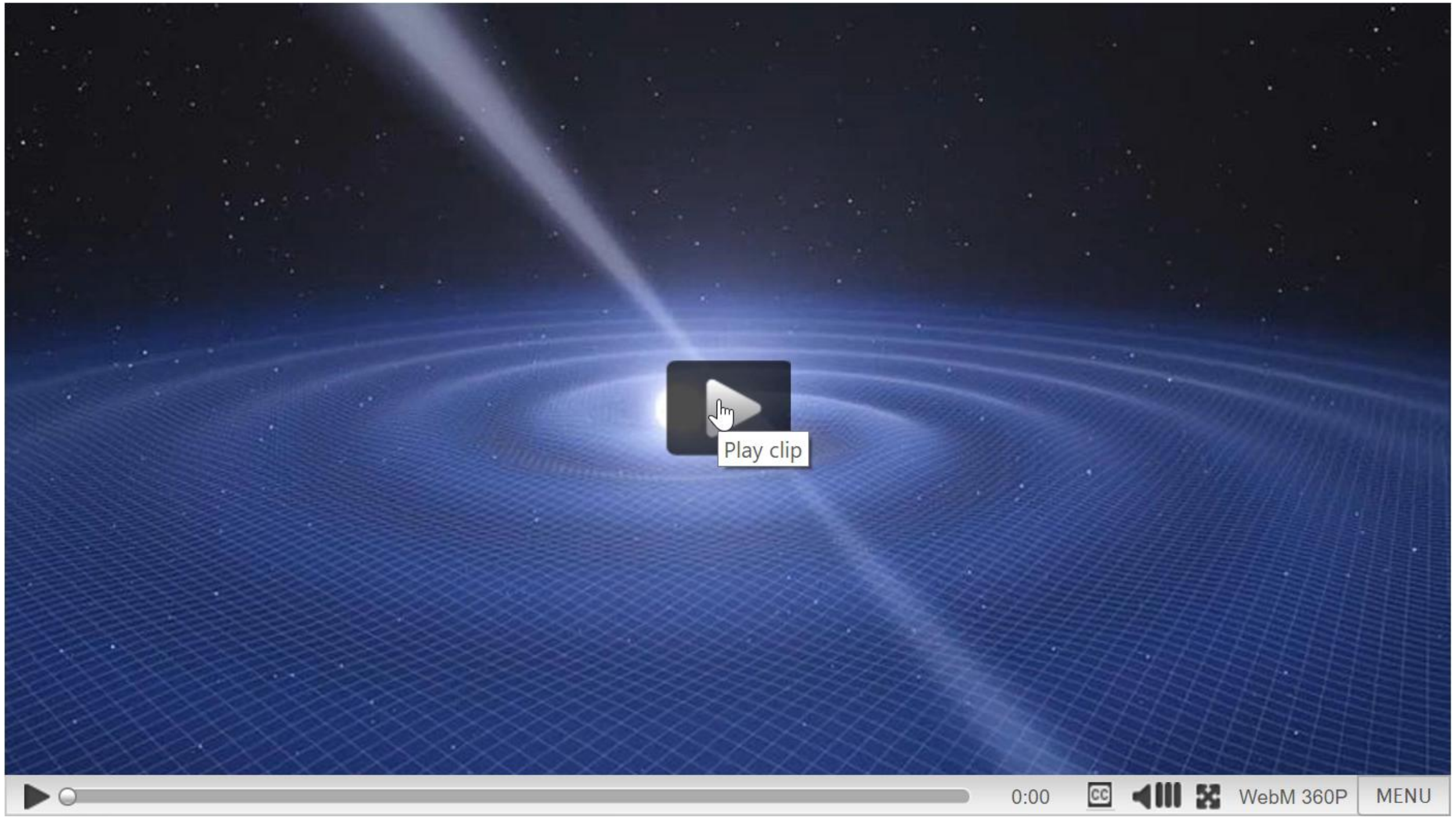
Sofia Dimoudi, Karel Adámek, Jack White, Mike Giles
(Oxford)



Light curve and slow motion picture of the **solitary** pulsar located in the centre of the Crab Nebula.

Image taken with a photon counting camera on the 80cm telescope of the Wendelstein Observatory, Dr. F. Fleischmann, 1998

Binary pulsars and gravitational waves



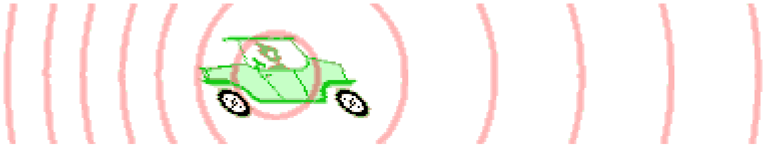
http://www.eso.org/public/videos/eso1319a/ Author: ESO/L. Calçada

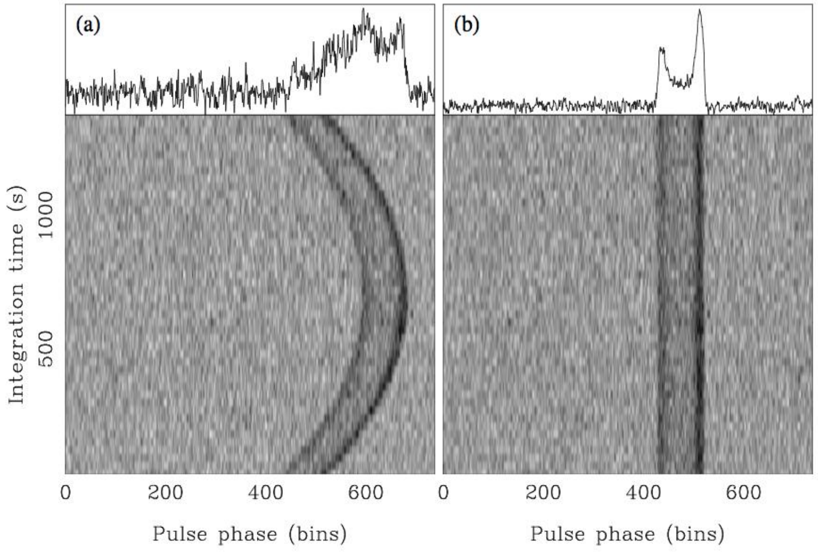
Fourier Domain Acceleration Search - FDAS

Signals from binary systems can undergo a **Doppler shift due to accelerated motion experienced over the orbital period**.

Much like the sound of a siren approaching you and then speeding away.

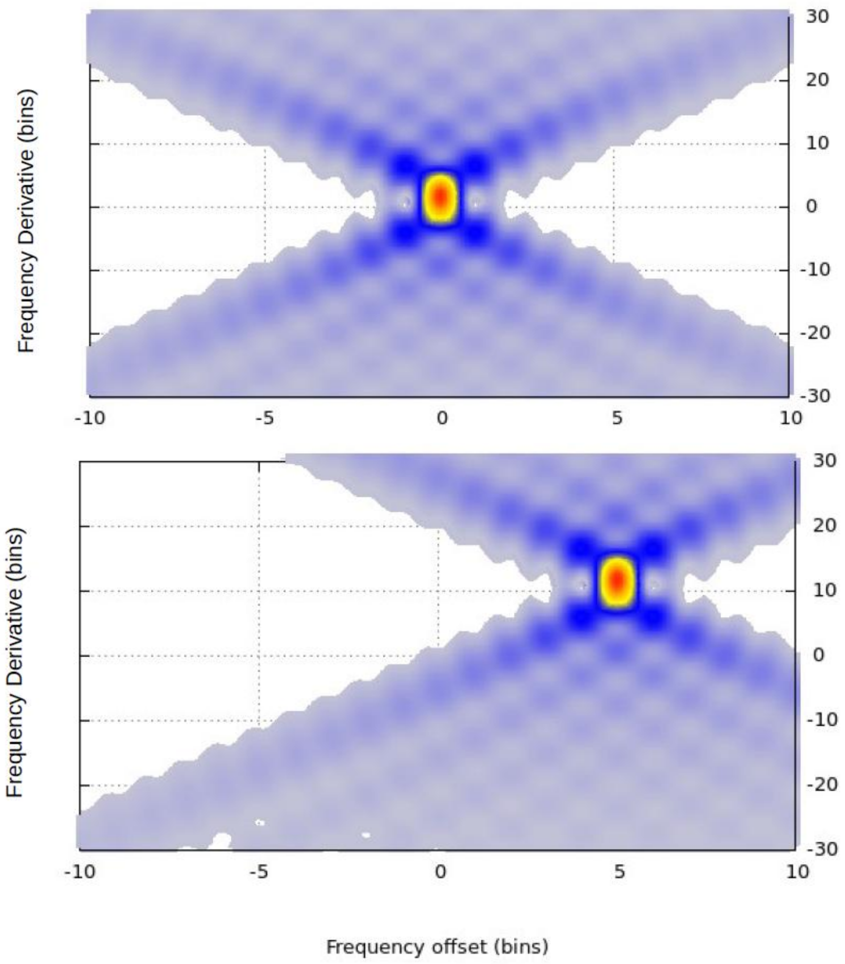
This can be corrected by using a matched filter approach.





Ransom, Eikenberry, Middleditch: AJ, Vol 24, Issue 3, pp. 1788-1809

FDAS Example



The two plots illustrate the effect of orbital acceleration.

The **first plot shows a signal without acceleration**, the signal is centred on its frequency and lies on the f-dot template corresponding to zero acceleration.

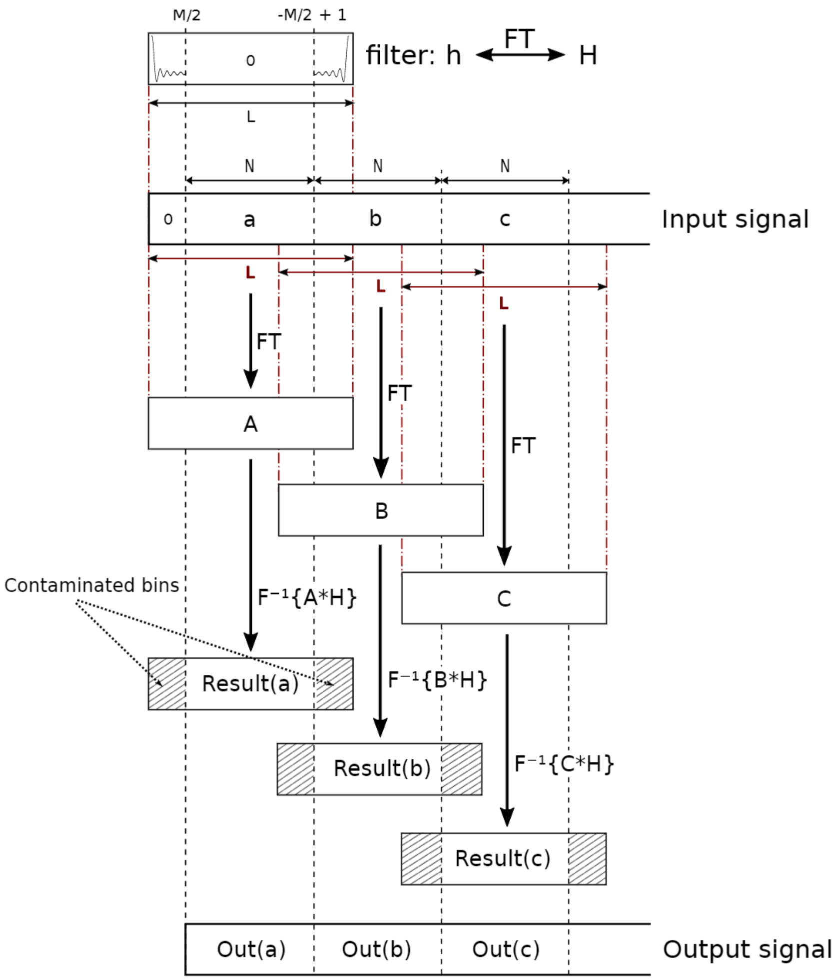
The **second plot shows a signal with a frequency derivative**, and has drifted from the original frequency by a number of bins.

Fourier Domain Acceleration Search

Use overlap-save algorithm to compute cyclic N-point convolution of template with signal segment.

Avoids the need for synchronisation because contaminated ends of convolved data are discarded (as opposed to overlap-add).

Code calculates the convolution, powers and extracts peaks.



GPU Fast Convolution via the Overlap-and-Save Method in Shared Memory: https://dl.acm.org/doi/10.1145/3394116

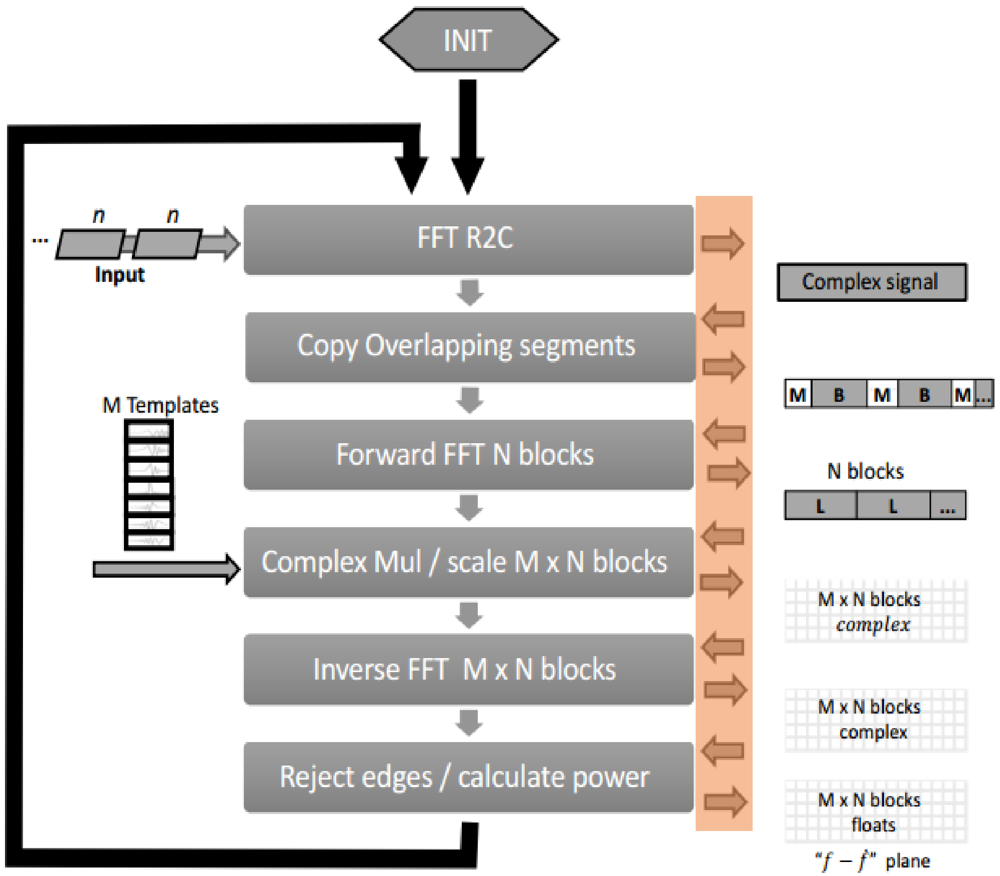
For more info: Kundur: https://www.comm.utoronto.ca/~dkundur/course_info/real-time-DSP/notes/8_Kundur_Overlap_Save_Add.pdf

Fourier Domain Acceleration Search

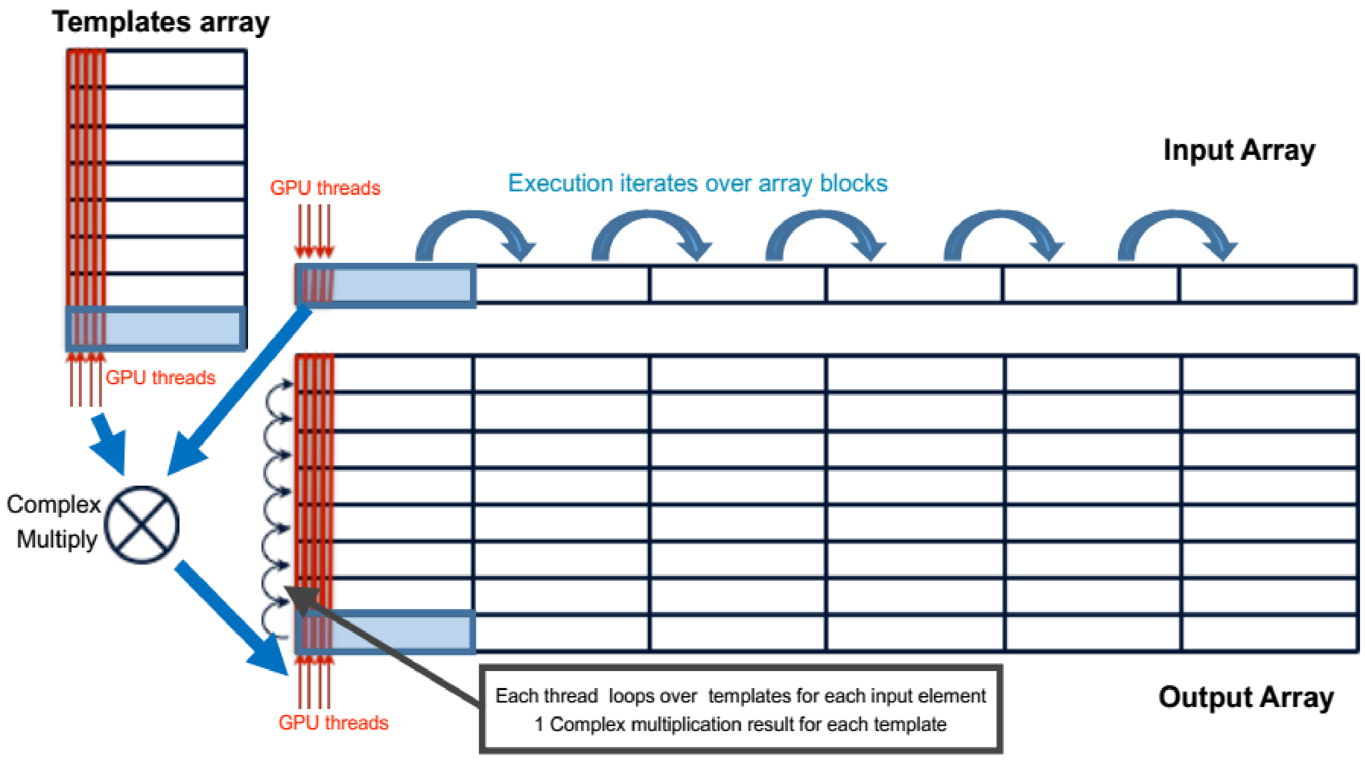
Using **cuFFT means many transactions to device memory on the GPU** (represented by grey arrows on the right of the diagram).

This causes the computation to be **limited by global memory bandwidth** (the lowest common denominator on a GPU).

This means that a cuFFT based implementation is very slow.



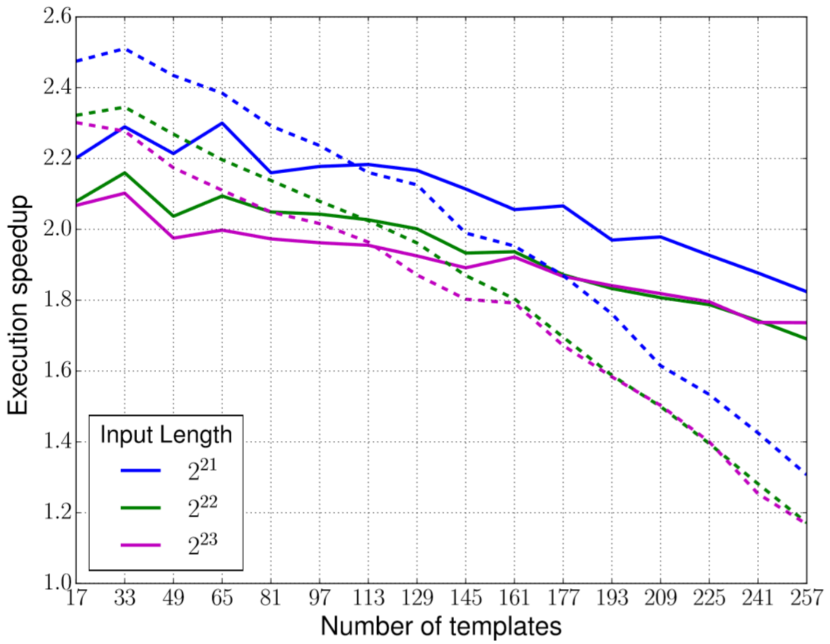
Fourier Domain Acceleration Search

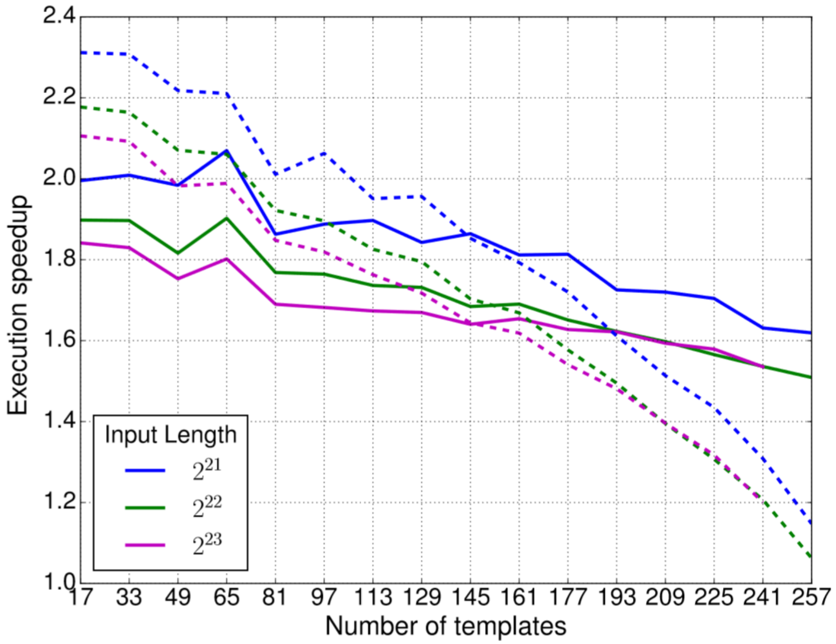


By writing our own custom I/FFT codes to work on shared memory we can perform the FFT, pointwise multiply and scale, IFFT and edge rejection all in one kernel.

Karel Adámek, Sofia Dimoudi, Mike Giles, and Wesley Armour. 2020. GPU Fast Convolution via the Overlap-and-Save Method in Shared Memory. *ACM Trans. Archit. Code Optim.* 17, 3, Article 18 (August 2020). https://doi.org/10.1145/3394116

Fourier Domain Acceleration Search





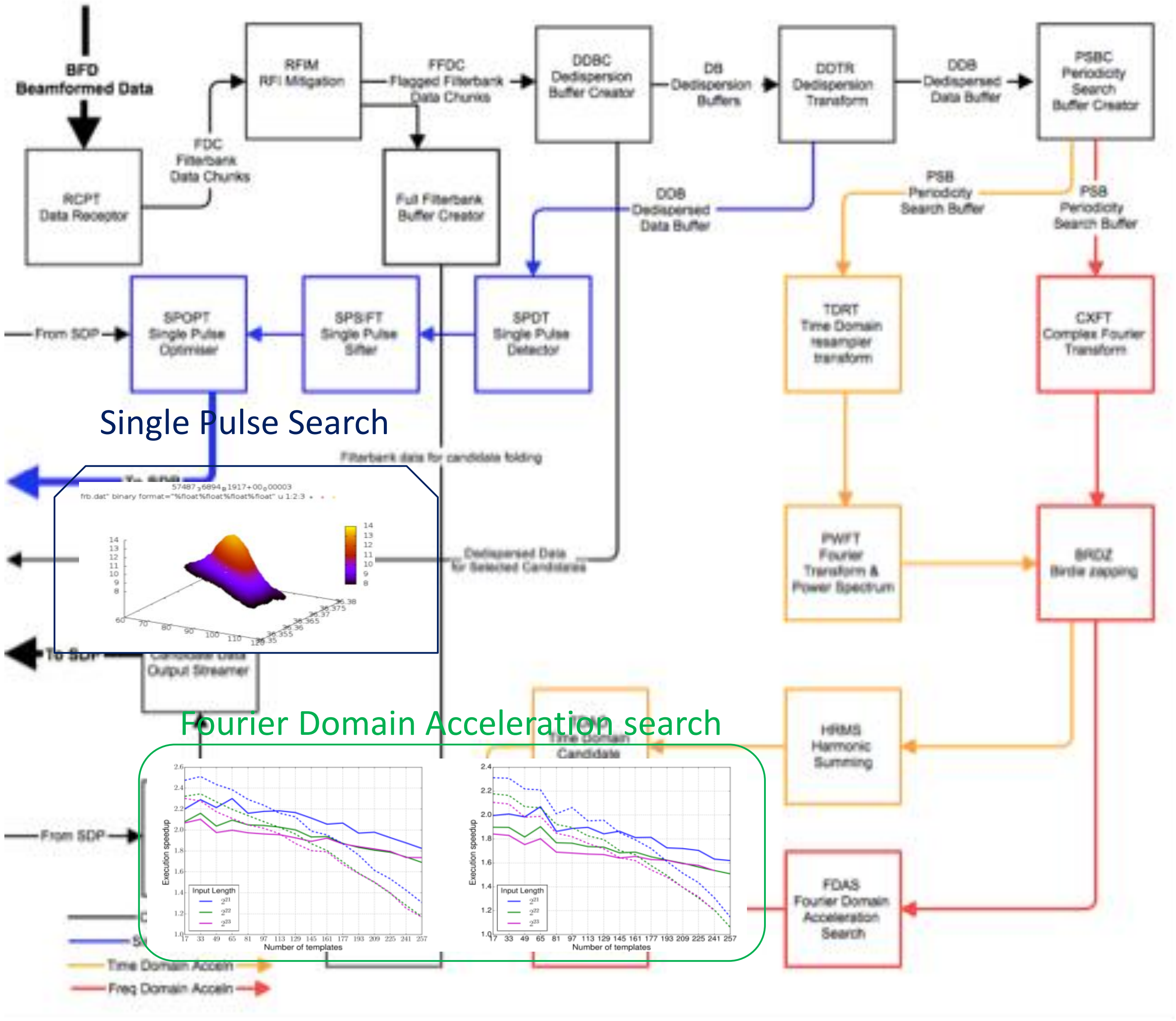
Initial results from our tests on a Tesla P100. In the SKA region of interest – signal length 2^{23}, template size of 512 (solid line) and no interbinning (left graph)

Further optimisation achieved approximately a 3.5x speed increase on a V100

Sofia Dimoudi *et al* 2018 *ApJS* **239** 28 https://iopscience.iop.org/article/10.3847/1538-4365/aabe88

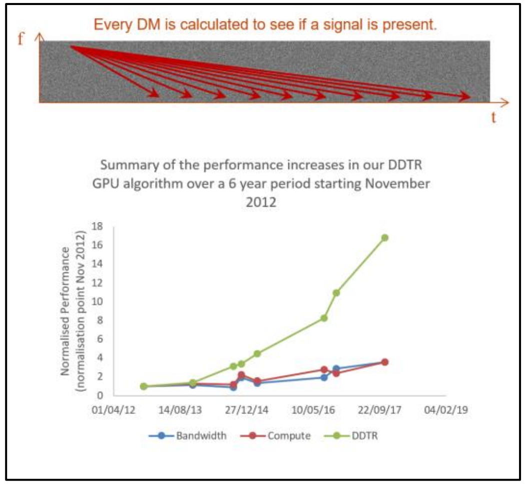
Karel Adámek, Sofia Dimoudi, Mike Giles, and Wesley Armour. 2020. GPU Fast Convolution via the Overlap-and-Save Method in Shared Memory. *ACM Trans. Archit. Code Optim.* 17, 3, Article 18 (August 2020). **https://doi.org/10.1145/3394116**

AstroAccelerate - Signal Processing



Single Pulse Search

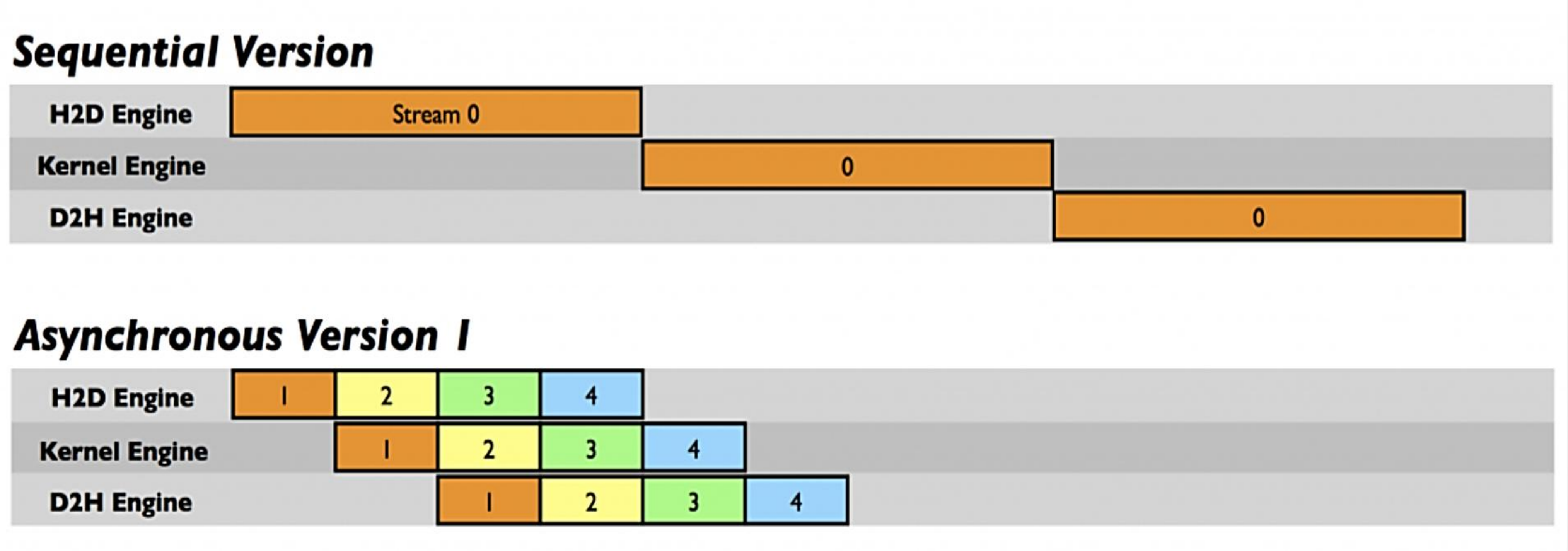
Fourier Domain Acceleration search



De-dispersion

Example use - AstroAccelerate

Use multiple streams to increase performance by overlapping memory communication with compute:

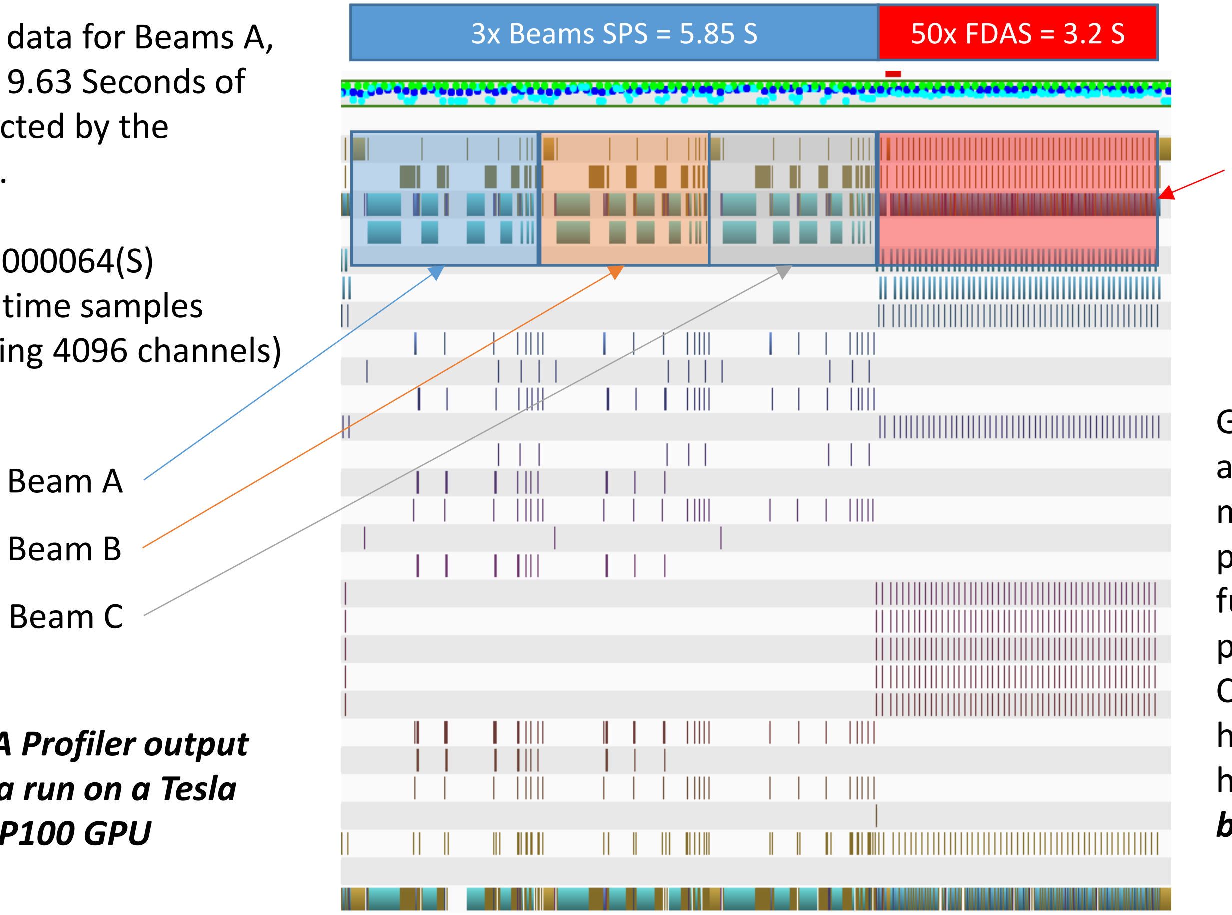


Prospects for SKA

3x Beams of SPS using 9.63S input chucks and 50 FDAS trials.
Total Processing taking about 9 seconds -> Faster than real-time.

The input data for Beams A, B and C is 9.63 Seconds of data collected by the telescope.

9.63(S)/0.000064(S)
= 150500 time samples
(each having 4096 channels)



NVIDIA Profiler output from a run on a Tesla P100 GPU

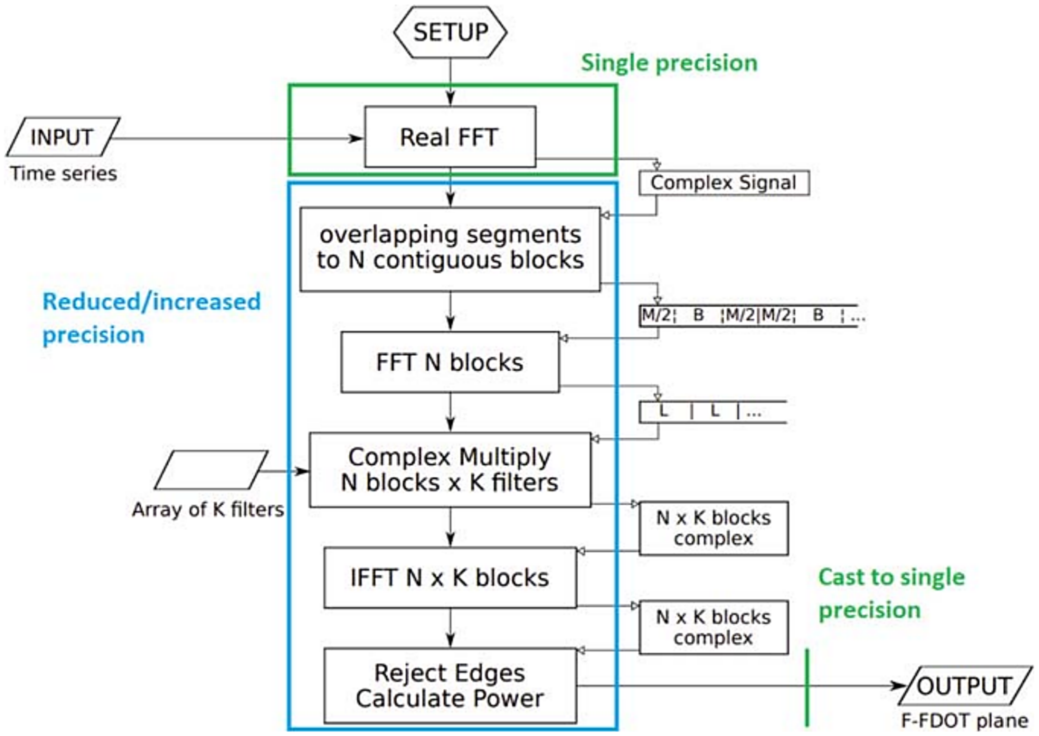
This shows 50 full resolution FDAS trials being performed from a previous pointing: 2^23 samples using 96 templates.
NO HARMONIC SUM

Given that each observation is about 536 seconds long this means that it is possible to perform 536/9.7 x 50 = 2750 full resolution FDAS trials while performing SPS on 3x beams. Current work indicates that the harmonic sum will (at most) half this -> ***450 FDAS trials per beam***

FDAS further optimisation

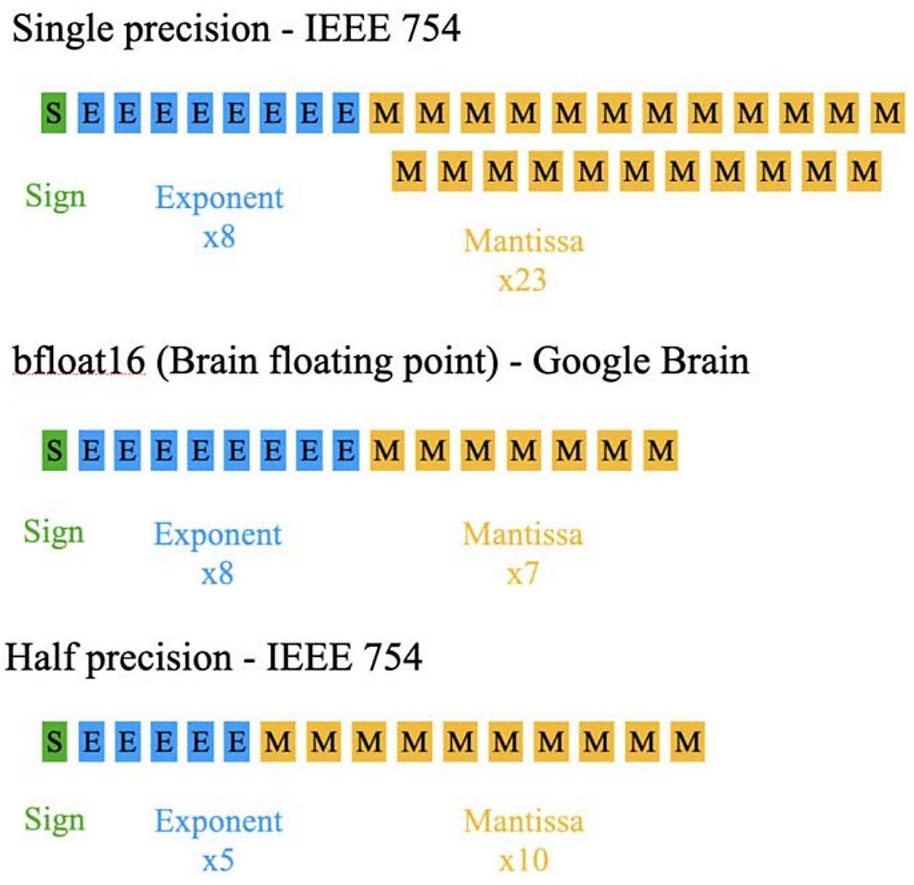
Jack's work has focused on implementing **mixed precision for FDAS**.

By reducing the precision in the convolution part of the algorithm (where FFTs can be shorter and we have far smaller accumulated errors) we are able to double the bandwidth throughput for the convolution parts of the code.



FDAS further optimisation

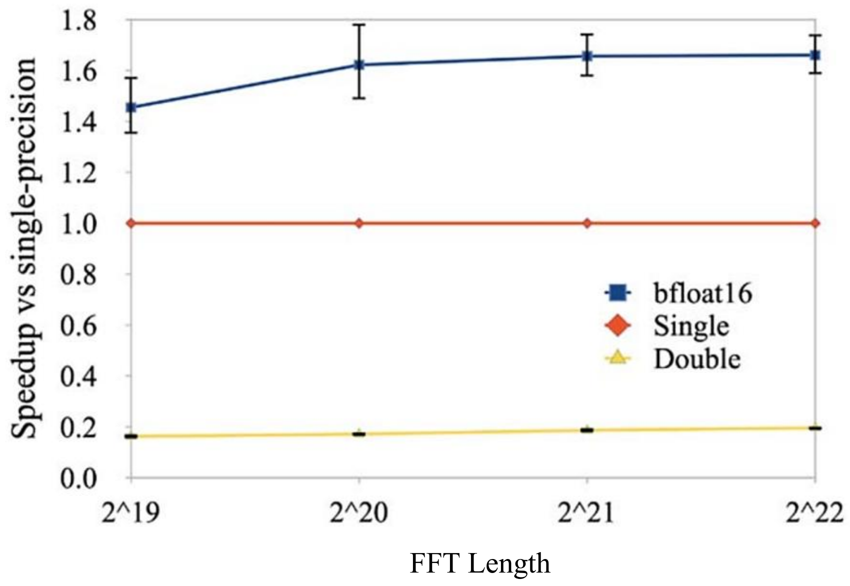
Jack used **bfloat16 (16 bits) for the convolution part of FDAS**. This is because it has the same range as float (fp32) so there is no need to scale numbers, and due to the fact that the exponent is the same as fp32 we are easily able to type convert between the two (on GPUs the SFU performs type conversion and it is a limited resource).



FDAS further optimisation

The work provides roughly a 1.7x speed increase compared to a single precision implementation.

So do consider whether you really need to compute in double or float or…..



Jack White *et al* 2023 *ApJS* **265** 13 https://iopscience.iop.org/article/10.3847/1538-4365/acb351/meta

FDAS Energy optimisation

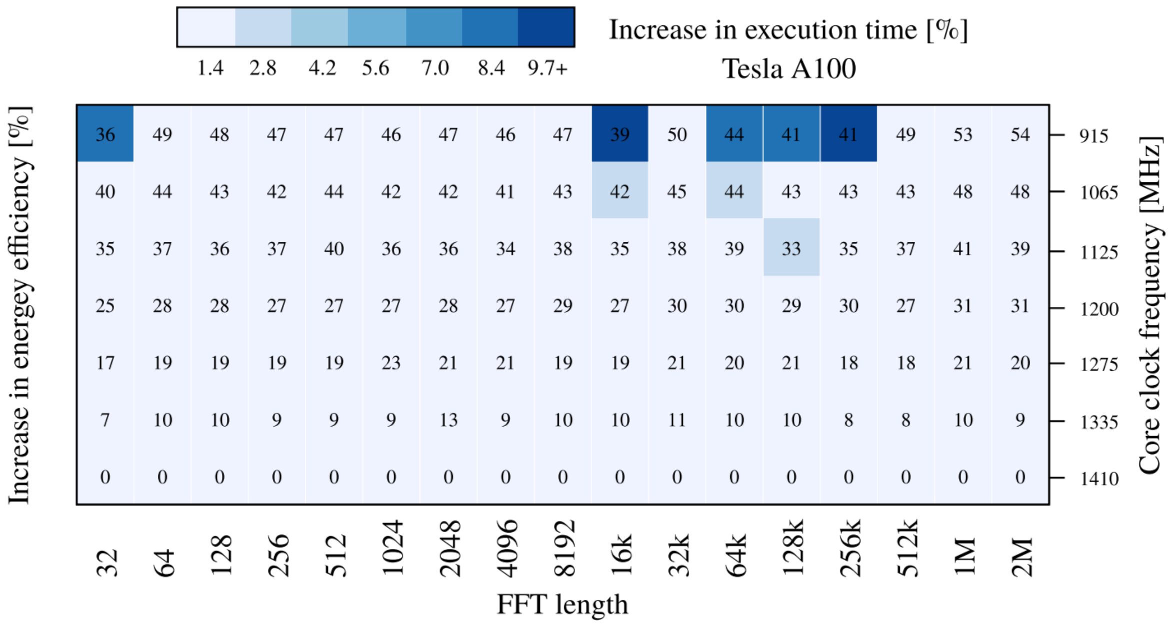
For a bandwidth bound algorithm it makes no sense to have the GPU cores running as quickly as they can.

The reason is that your memory cannot deliver data to them quickly enough to utilise all of the computation that they can perform.

FDAS Energy optimisation

Because the **FFT is bandwidth bound we are able to reduce the GPU core clock frequency without having significant impact on execution time.**

The figure (right) shows the energy saving for a given core clock frequency. The shading indicates the increase in execution time.



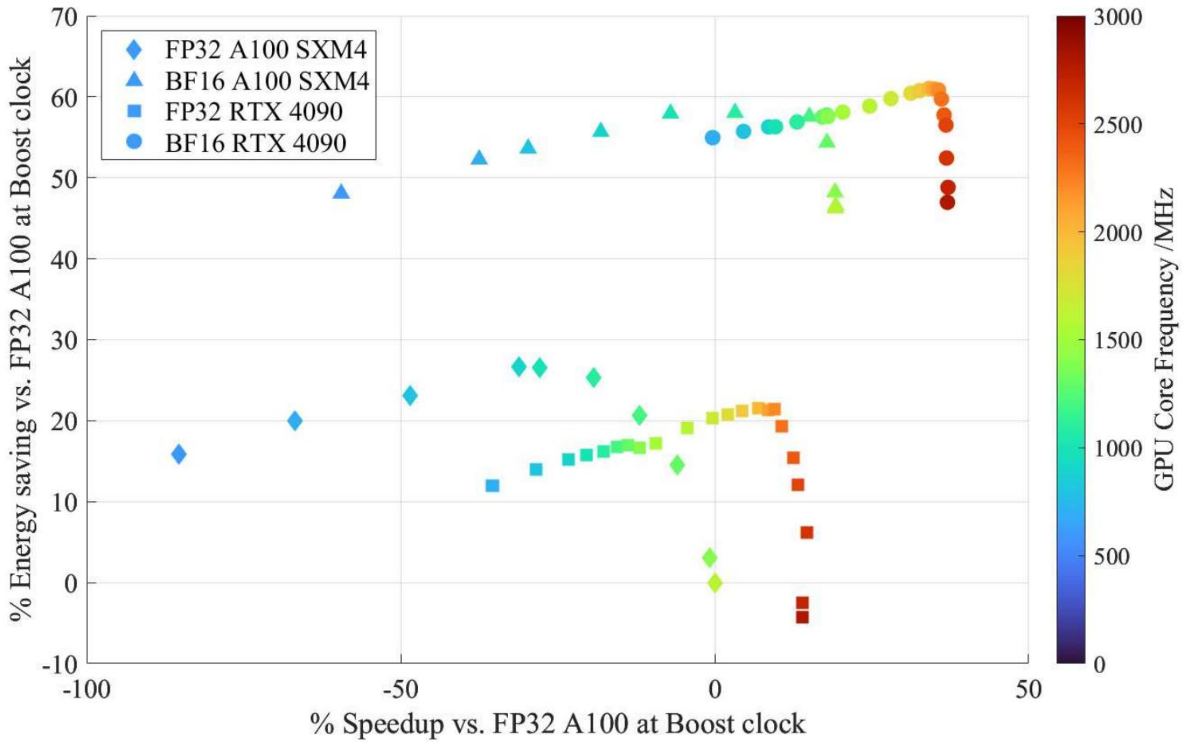
K. Adámek, et al in *IEEE Access* https://ieeexplore.ieee.org/abstract/document/9330509

FDAS Energy optimisation

We can use NVML to change the clock speed of the processing cores to slow them down.

This has the advantage of reducing the energy that the GPU uses.

By combining mixed precision with reduction in core clock frequency we are able to achieve speed increase and energy savings.



Jack White *et al* Astronomical Data Analysis Software and Systems, ADASS XXXII

A final word on measurement

Often nvidia-smi is used to establish values of physical parameters of the GPU whilst it is executing code.

If you have workloads with short duration kernels this might not be the best thing to do…

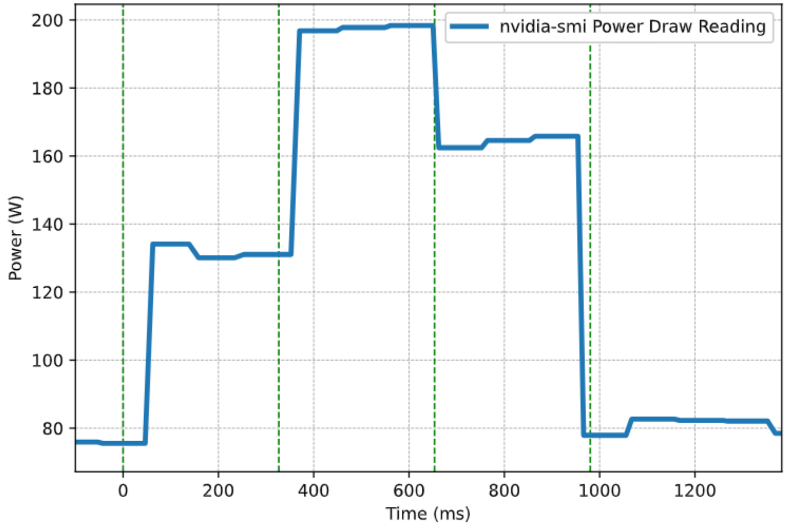
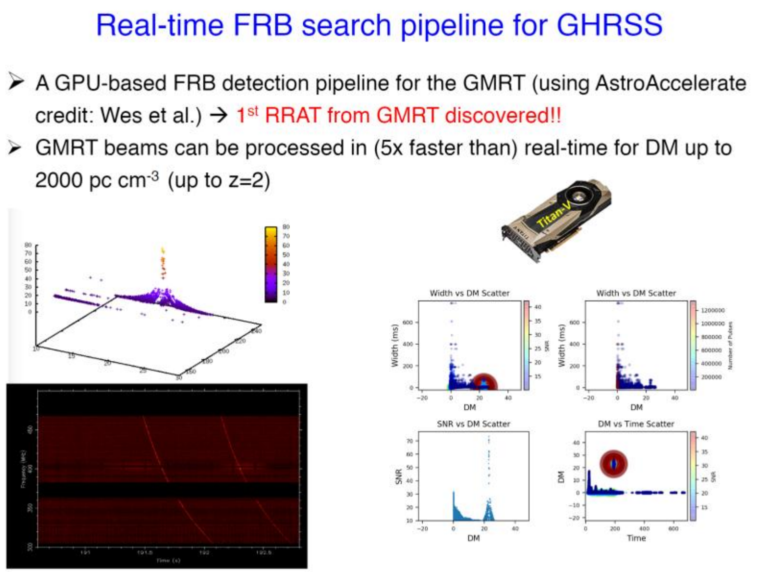


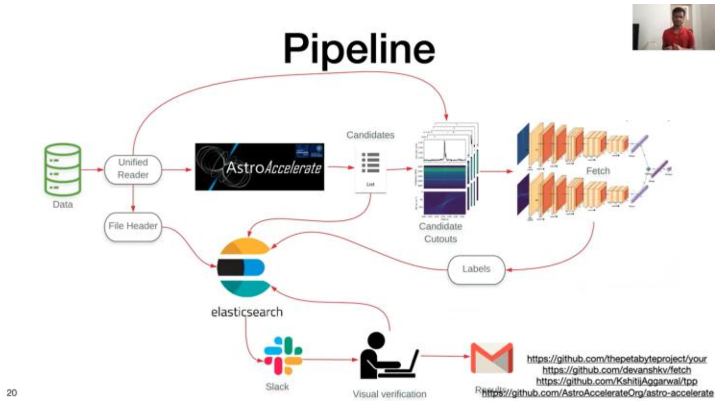
Figure 1: nvidia-smi can report drastically different power draw for the same CUDA Kernel, ranging from 80W to 200W. This is because nvidia-smi does not fully capture the power information on some GPUs. The figure shows a CUDA program that runs for 325ms on an A100 GPU. The kernel is executed 4 times, and the green dotted line indicates the beginning of each iteration.

Jim Yang et. al. (to appear in SC24 proceedings) https://arxiv.org/abs/2312.02741

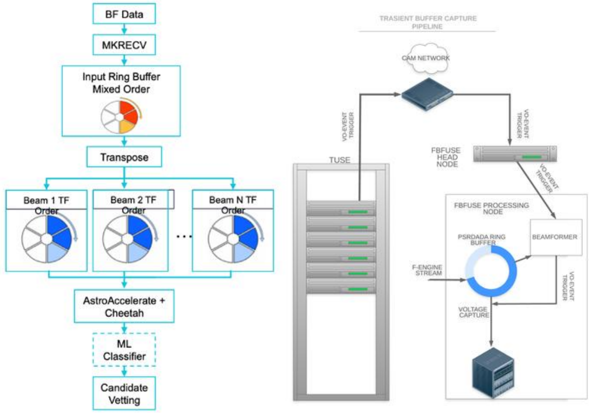
Impact

Giant Metrewave Radio Telescope
[Presentation]





The Petabyte FRB Search Project
[Presentation]



MeerKAT (MeerTRAP)
[paper] ([arxiv])

Acknowledgments and Collaborators

University of Oxford

Mike Giles (Maths)

Aris Karastergiou (Physics)

Chris Williams (Physics)

Steve Roberts (Engineering)

Sofia Dimoudi (OeRC)

Nassim Ouannoughi (OeRC)

Karel Adamek (OeRC)

Jayanth Chennamangalam (Physics)

Griffin Foster (Physics)

University of Manchester

Ben Stappers

Mike Keith

Prabu Thiagaraj

Jayanta Roy

Mitch Mickaliger

NRAO/UVA

Scott Ransom

University of Bristol

Dan Curran (Electrical Engineering)

Simon McIntosh Smith (Electrical Engineering)

ASTRON

Cees Bassa

Jason Hessels

University of Opava

Jan Novotny

NVIDIA

Kate Clark

Tim Lanfear

Tom Bradley

Max Plank

Ewan Barr

http://www.oerc.ox.ac.uk/projects/astroaccelerate