Fourier Domain Pulsar Acceleration Searches on GPUs for the Square Kilometre Array

Sofia Dimoudi[1] and Wes Armour[1]
1. Oxford e-Research Centre, University of Oxford
sofia.dimoudi@oerc.ox.ac.uk

Fourier domain Matched Filtering

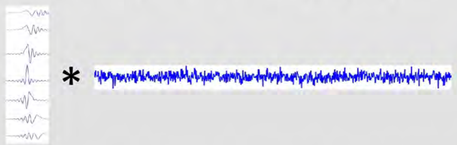
Correlation Technique

Orbital acceleration is expressed in a number of FIR filter templates. Correlating signal with the inverse filters corrects the effect (S. Ransom, 2002[1]).

$$A_{r_0} = \sum_{k=r_0-m/2}^{r_0+m/2} A_k A^*_{r_0-k}$$

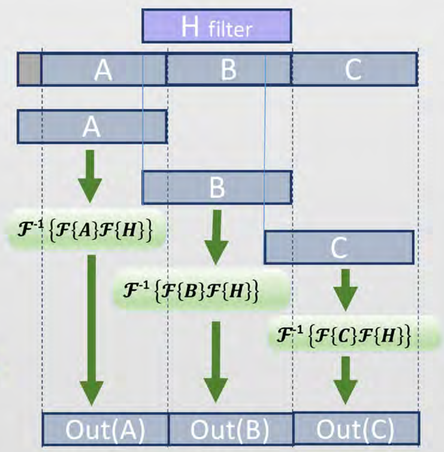
Achieved easily using convolution in the Fourier Domain:

$$\mathcal{F}\{A_{r_0}\} = \mathcal{F}\{A_k\}\mathcal{F}\{A^*_{r_0-k}\}$$



Overlap-save method

- Short filters ($m < 300$)
- long signal (N = 2^{22} fourier points)
- Computed more efficiently in blocks
  - Easily parallelizable
  - Edge removal required

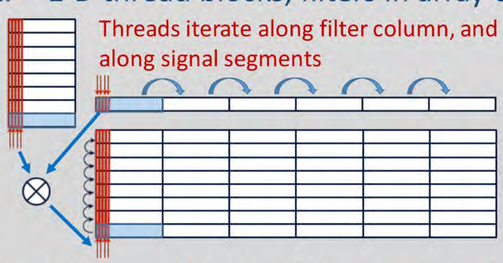


Paralellizing multiple block convolutions

- 6000 inputs, 2^{22} points each, 96 filter templates
- Each template convolves with signal segments iteratively and the result fills an "$f - \dot{f}$" matrix in global memory
- Computational grid is at the template size level

1. 1-D thread blocks, filters in array of registers

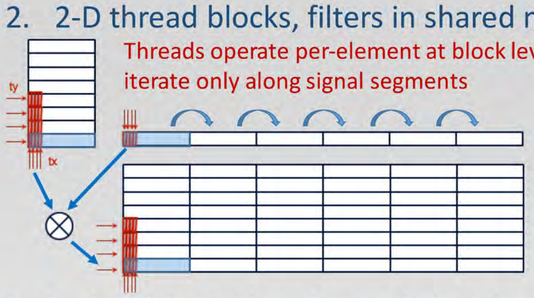
Threads iterate along filter column, and along signal segments



- Register pressure
- Loop unrolling gives x 1.7 performance (no register spills, low occupancy)
- Use of __ldg() intrinsic for faster global loads

2. 2-D thread blocks, filters in shared memory

Threads operate per-element at block level, iterate only along signal segments

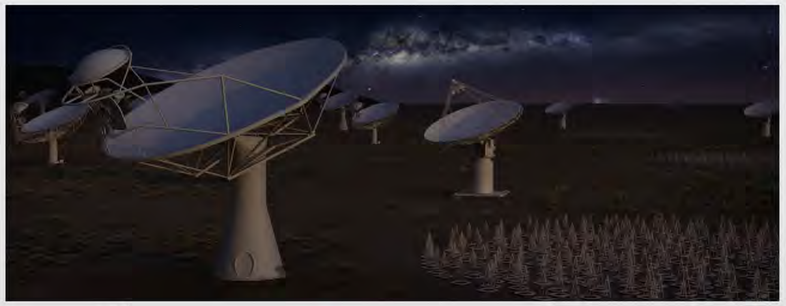


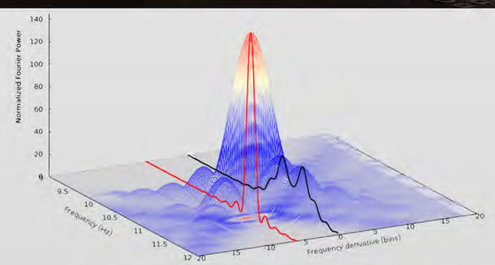
- Limited by maximum threads
- 100% occupancy
- Best performing kernel

➢ Bandwidth limited kernels, convolution schemes restricted by the use of cuFFT

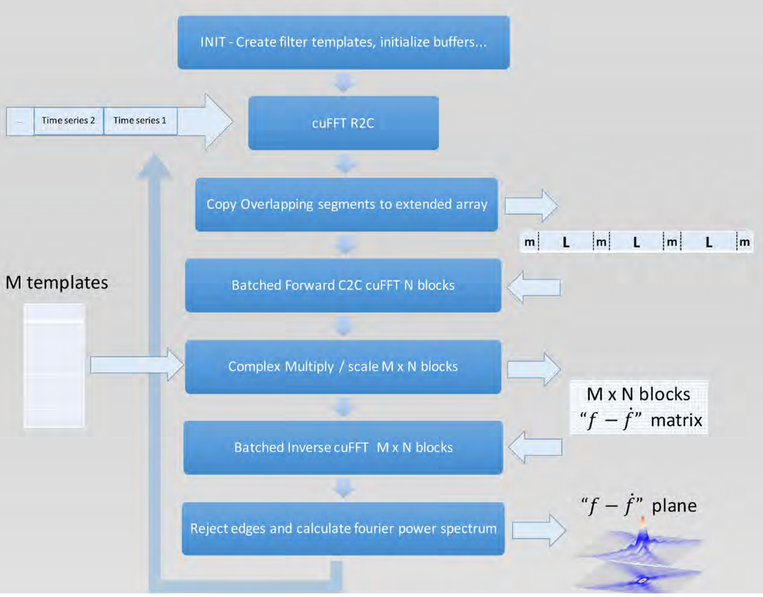
Pulsar Acceleration Searches

Pulsar acceleration searches are methods for recovering signals on radio telescopes that are affected by orbital acceleration in binary systems. The vast amount of data that will be produced by next generation instruments such as the Square Kilometre Array (SKA) necessitates real-time operation, which in turn requires the use of HPC platforms. An implementation of the Fourier Domain Acceleration Search (FDAS) algorithm on GPUs in the context of the SKA, is currently under development at the Oxford e-Research Centre (OeRC), University of Oxford, as part of the Astro-Accelerate real-time data processing library.

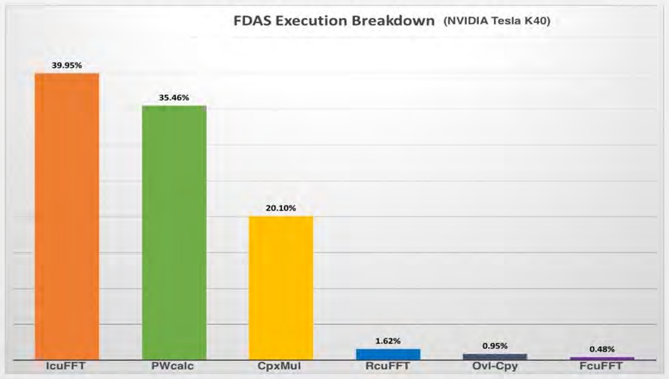




Data Flow

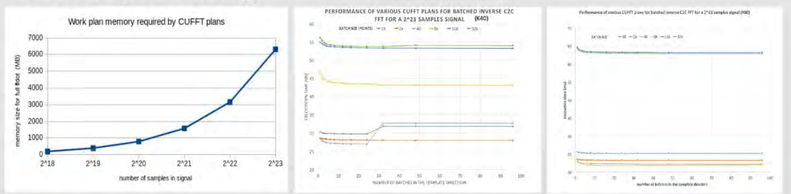


Performance



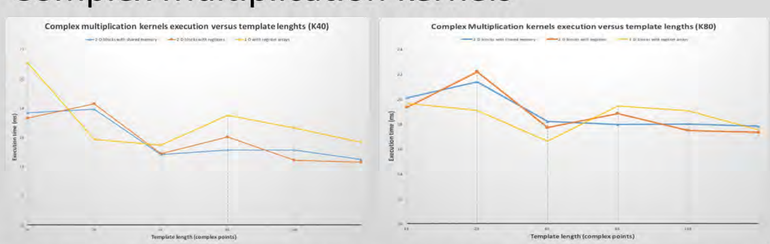
- Computations are dominated by the M X N blocks inverse FFT and the power operations
- Significant optimization possible by eliminating data transfers between cuFFT calls

cuFFT planning



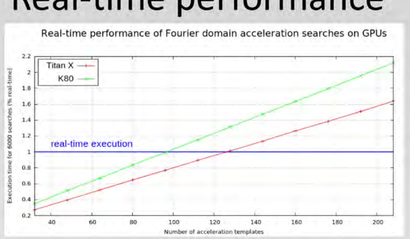
Plan size balance: cuFFT static plans require temporary storage, sequential execution of smaller plans may take longer.
K40 performs best for the full data set with 4Kpts batch size, K80 with 8Kpts.
Execution time does not improve for plans with more than 24 x 1024 batches.

Complex Multiplication kernels



The register intensive kernel performs better on K80 at the desired sizes, benefiting from the double register file.

Real-time performance



The algorithm is achieving so far better than real-time performance, on a single GPU for the target specifications.