Signal detection and data processing for LOFAR and SKA using GPUs

Wes Armour
James Martin Fellow
Oxford e-Research Centre

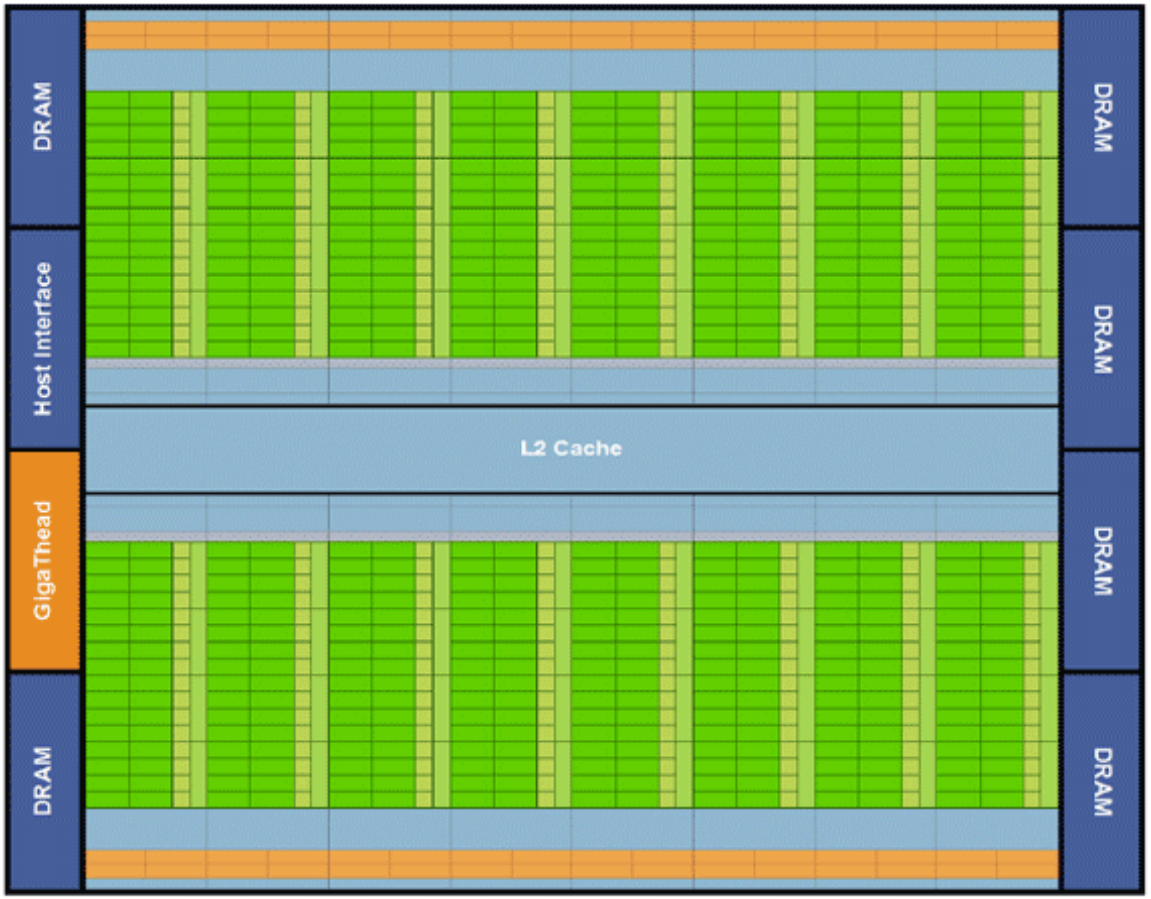
20th May 2011

Current status of GPUs

Overview of current GPUs - Fermi

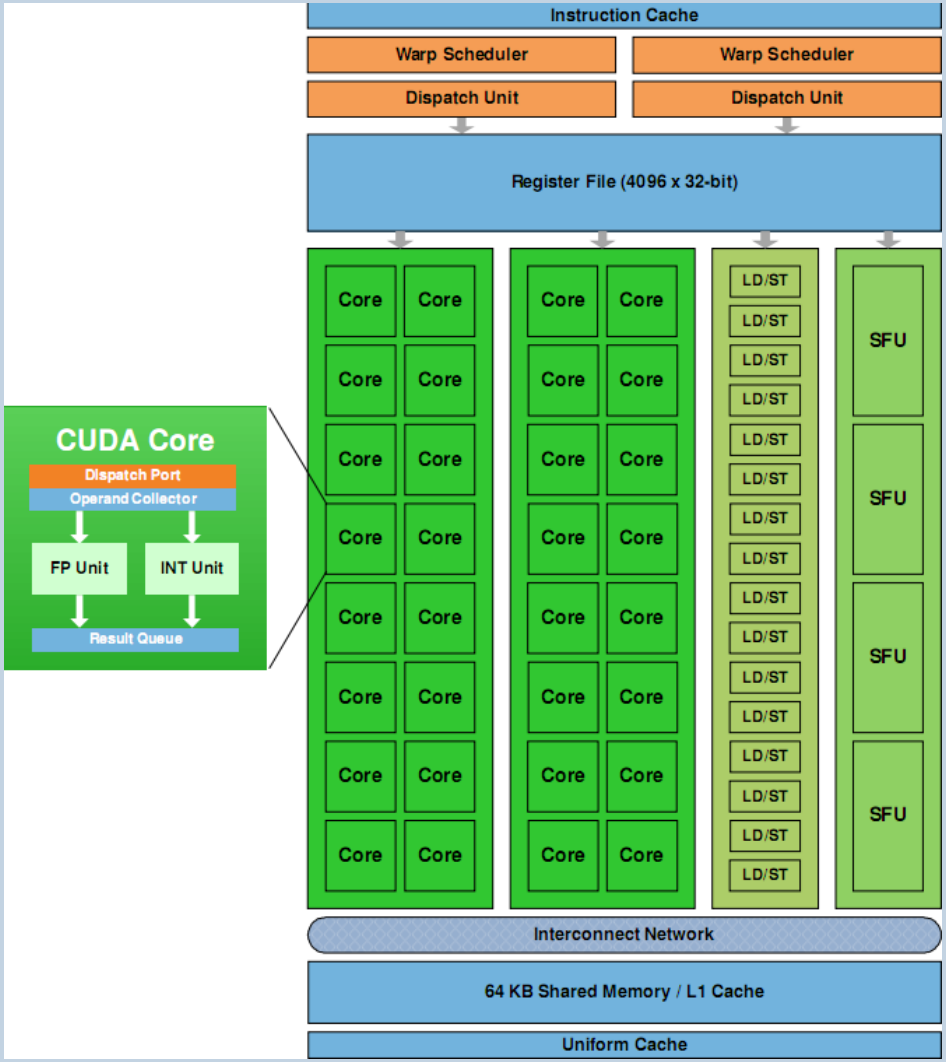
- Fermi released 2010
- 32 cores per Streaming Multiprocessor (SM)
- Configurable 16/48K or 48/16K L1 cache / shared memory (per SM)
- New L2 cache – 768K
- GigaThread – Concurrent kernel execution (16)
- ECC – slows things down, 10% - 25%
- Simple to use extensions - CUDA for C, C++ & FOTRAN

Fermi Design



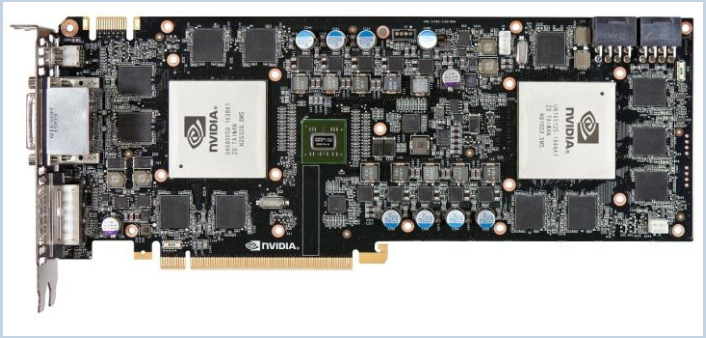
Fermi's 16 SM are positioned around a common L2 cache. Each SM is a vertical rectangular strip that contain an orange portion (scheduler and dispatch), a green portion (execution units), and light blue portions (register file and L1 cache).

Fermi Streaming Multiprocessor (SM) design



Latest Fermi based cards

- GeForce GTX 590 – 2x GF110 GPUs
- 6.8 GFLOPS / watt
- Price point ~ £600



- M2070 – 1x GF100 GPU, 16 SMs
- 5.7 GFLOPS / watt
- Price point ~ £2.5K

Influence and Take-up

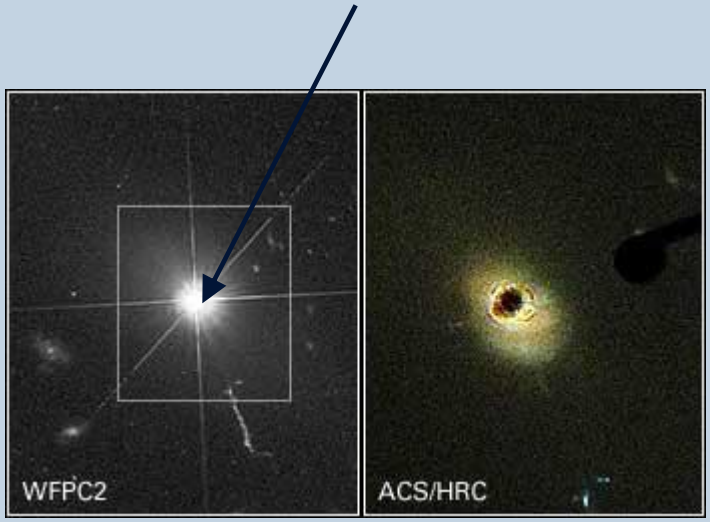
TOP 500 – 3 out of top 5 utilise Fermi/Tesla

- Tianhe-1A 2.5 petaflops
- Based on 14336 Xeon and 7168 M2050
- To achieve same performance using only CPUs 50000 CPUs, 2x floor space and 3x power (Estimates made by NVIDIA)
- Uses Lustre :-s

Dispersion and Dedispersion

Radio Astronomy and Radio Transients

Quasars – Energetic region of a distant galactic core, surrounding a supermassive black hole

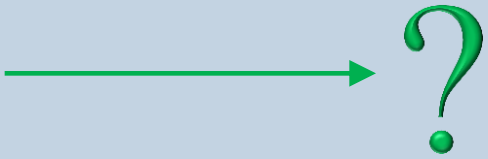


NASA and J. Bahcall (IAS)

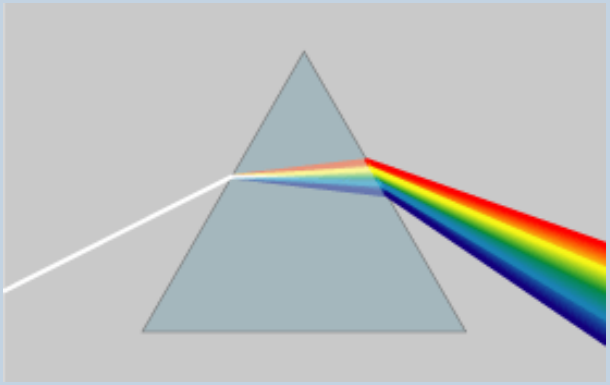
Hester et al.

Pulsars – Magnetized, rotating neutron stars. Emit synchrotron radiation from the poles, e.g. Crab Nebula

RRATS – Rotating Radio Transients. Short, bright irregular radio pulses. Discovered 2006

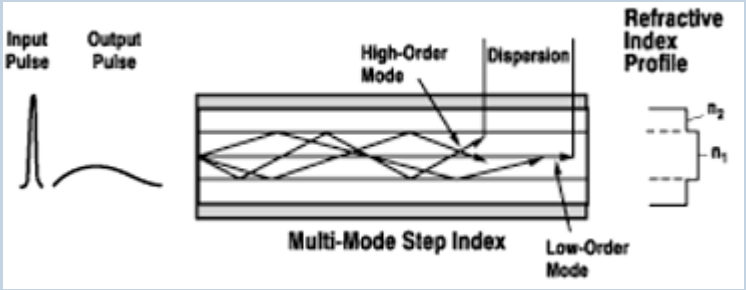


Dispersion



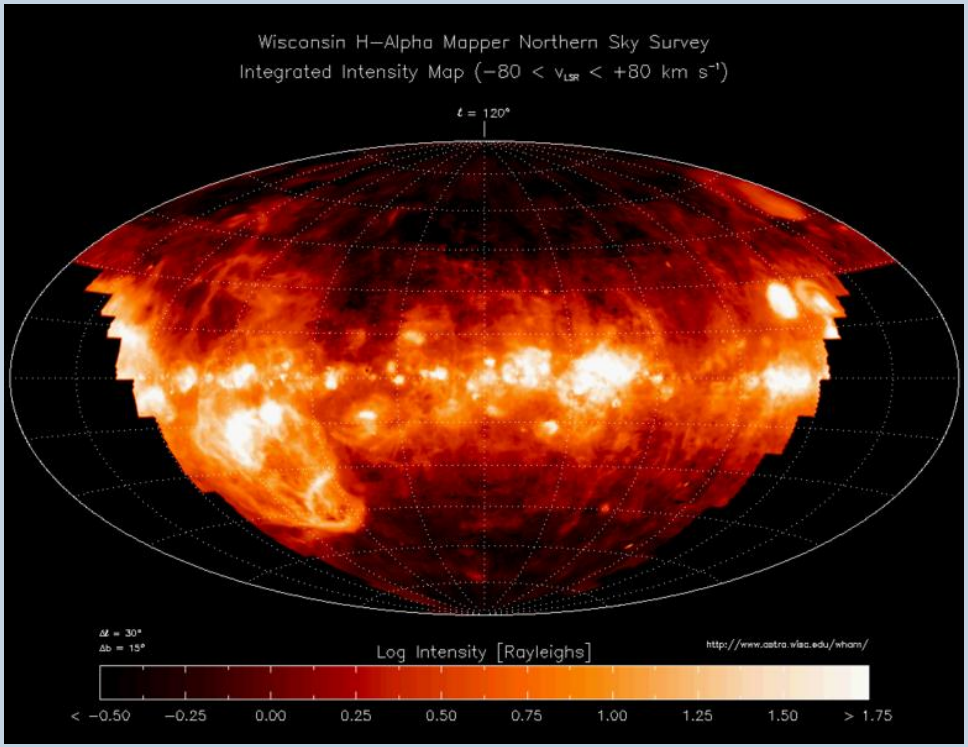
Chromatic dispersion is something we are all familiar with. A good example of this is when white light passes through a prism.

Group velocity dispersion occurs when pulse of light is spread in time due to its different frequency components travelling at different velocities. An example of this is when a pulse of light travels along an optical fibre.



Dispersion of Radio waves by the ISM

The interstellar medium (ISM) is the matter that exists between stars in a galaxy.



Haffner et al. 2003

In warm regions of the ISM (~8000K) electrons are free and so can interact with and effect radio waves that pass through it.

The Dispersion Measure - DM

The time delay, $\Delta\tau$, between the detection of frequency f_{high} and f_{low} is given by:

$$\Delta\tau = C_{DM} \times \mathrm{DM} \times \left(\frac{1}{f_{low}^2} - \frac{1}{f_{high}^2} \right)$$

Where C_{DM} is the dispersion constant. DM is the dispersion measure:

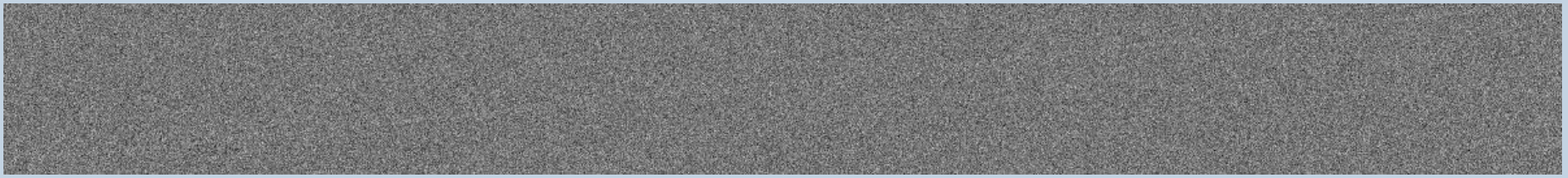
$$DM = \int_0^d n_e dl$$

This is the free electron column density between the radio source and observer.

We can measure $\Delta\tau$ and f and so can study DM

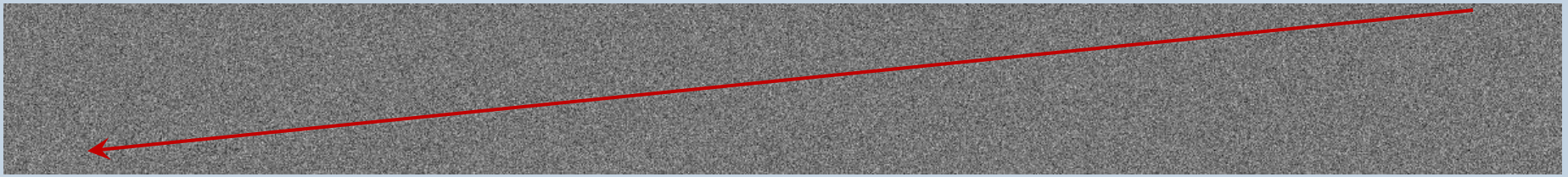
Experimental Data

Most of the measured signals live in the noise of the apparatus.

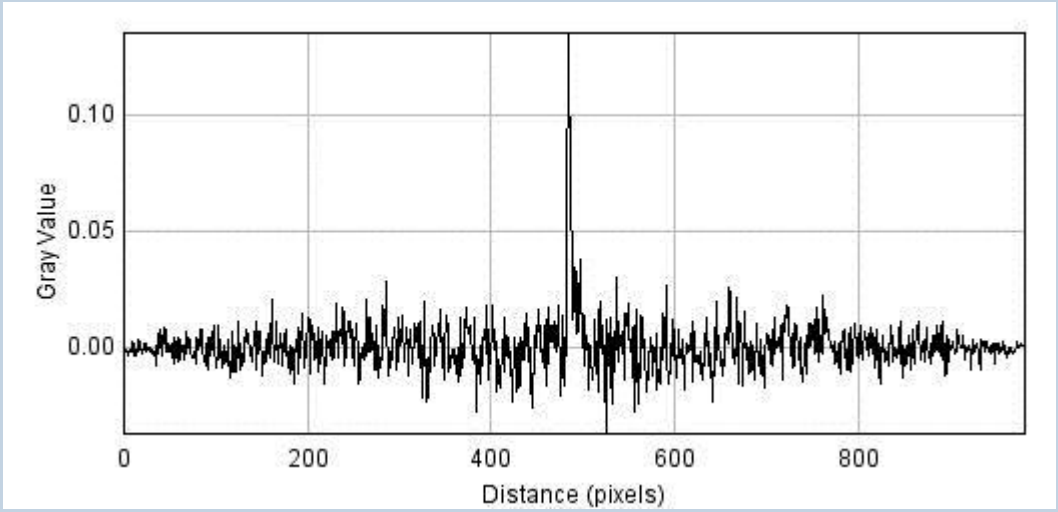


Experimental Data

Most of the measured signals live in the noise of the apparatus.



Hence frequency channels have to be “folded”



OXFORD e-Research CENTRE

MDSM – Alesseo, Brute force algorithm.

Algorithm 3.1 Brute-Force Dedispersion CUDA implementation

Require: nsamp, nchans, startdm, dmstep, tsamp, dm_shifts

$s \leftarrow threadId.x + blockIdx.x * blockDim.x$

$shift\_temp = {}^{startdm+blockIdx.y*dmstep}/_{tsamp}$

for $samp = 0$ to $nsamp$ **do**

$samp \leftarrow samp + blockDim.x * gridDim.x$

$soffset \leftarrow s + samp$

for $c = 0$ to $nchans$ **do**

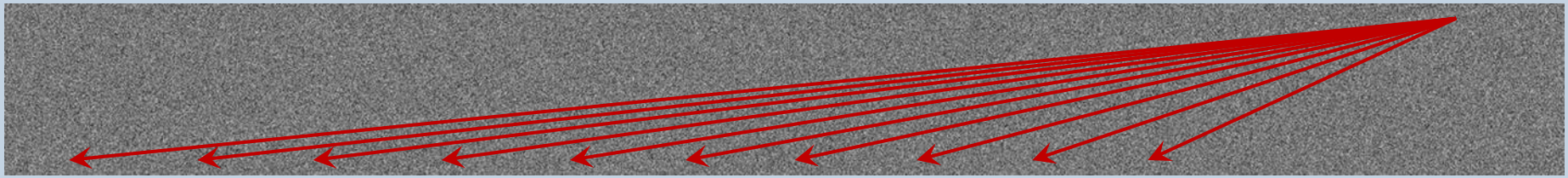
$index \leftarrow (soffset + dm\_shifts[c] * shift\_temp) * nchans * c$

$output[blockIdx.y * nsamp + soffset] += input[index]$

end for

end for

Every DM is calculated to see if a signal is present.



1st Stage – Memory access optimisation

- Begin by profiling the code using the Cuda Profiler.
- Identified bottle necks in memory access patterns
- By moving the variable that holds the running total from shared memory to a register kernel execution was 5x faster.

2nd Stage – Occupancy Optimisation

- Optimize GPU block size for each kernel to get the maximum occupancy of threads on the GPU
- Investigated the optimal block size for GPU hardware versions sm_13 and sm_20
- This produced a 1.25x speed increase.

Current Speedup

On-going and Future Work

- Mikes new algorithm for GPU

Acknowledgments