A Software Correlator for SKA$_1$

Dominic Ford, Andy Faulkner & Paul Alexander

Introduction

Cross correlators are a highly computationally intensive part of any radio interferometer. Although the number of operations per data sample is small, correlation occurs before time integration and so the sample rate is very high – it must be at least twice the radio-frequency bandwidth of the telescope.

At present, FPGA and/or ASIC based correlators are widely used. These offer highly transistor- and power-efficient solutions as logic units are hard-wired to solve the particular correlation problem required. However, this efficiency comes at the expense of flexibility.

Software correlators, running on general-purpose CPUs or GPUs offer the ability to easily add antennas, change the number of baselines correlated, or tune the spectral resolution of the telescope as required. The specification of the telescope can be scaled with time as more computing power becomes available.

Several implementations already exist. LOFAR uses an IBM Bluegene supercomputer as its correlator. An MPI-based correlator for use on x86 compute clusters has been developed by Deller *et al*. (2007) at Swinburne and is in use at the LBA. In August 2011, Clark *et al*. released the *Harvard X-Engine* – a GPU cross-correlation code which achieves nearly 80% core utilisation on NVIDIA GPUs.

LOFAR uses an IBM Bluegene/P as its correlator (Romein *et al*. 2010).

The 32T MWA tested a GPU-based correlator (Wayth *et al*. 2007), and the 128T will use a hybrid FPGA/GPU correlator.

Imagine the SKA with a software correlator

✓ **Rapid development cycle**.

Software correlators are designed using conventional programming languages for which compilers and debugging suites already exist.

✓ **Reduced NRE**.

Hardware is already well tested and mass-produced.

✓ **Deploy the correlator in phases.**

✓ **Track hardware developments**.

Short development cycles mean that hardware can be deployed in the field within months of being released.

✓ **Easy upgrade path**.

After four years, it may save money to replace the correlator with new more power-efficient hardware. The cost of software re-development is minimal compared to the cost of ASIC/FPGA development.

✓ **Easy reconfigurability**.

Switch to new algorithms post-deployment and explore new approaches which integrate with the post-processor and scale to Phase 2.

✓ **Excellent fault tolerance**.

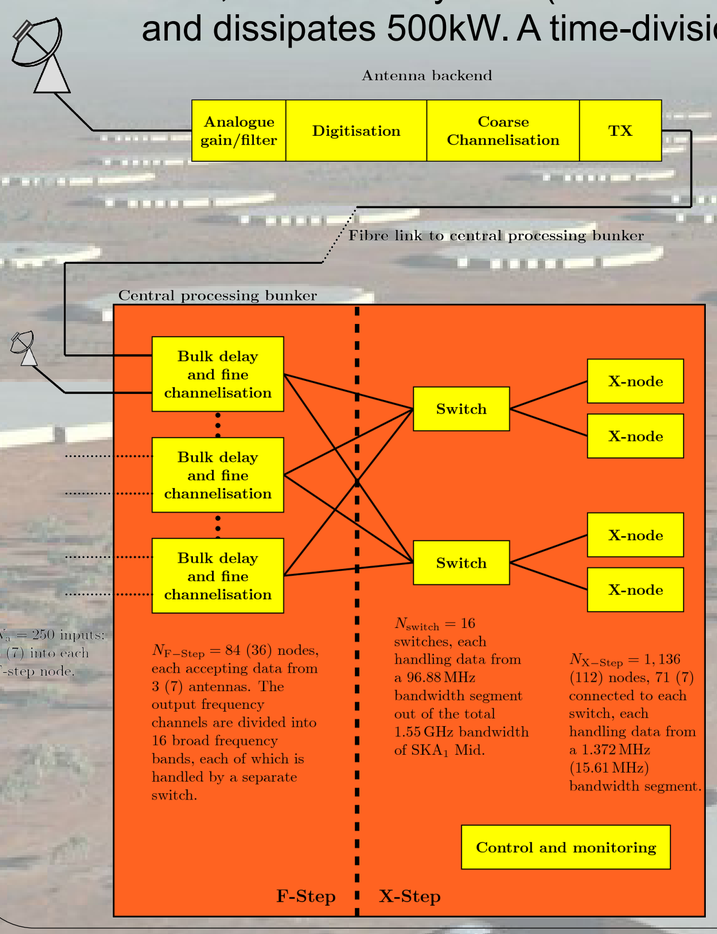
Respond on-the-fly to hardware or antenna failure.

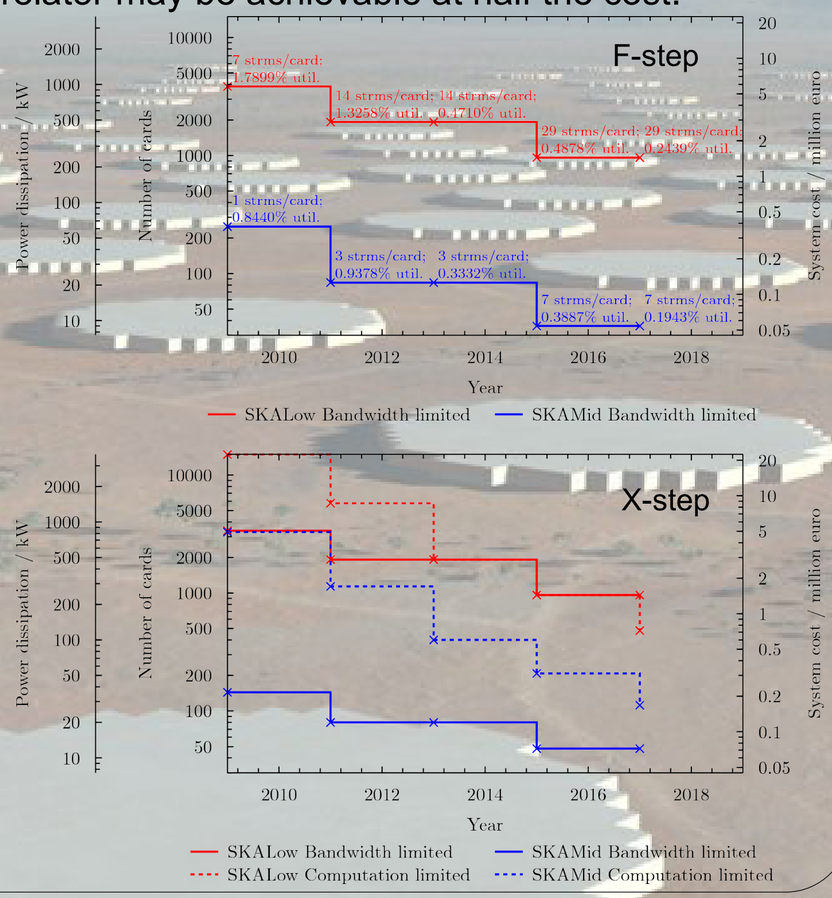
✓ **Share hardware**.

Why buy a separate beamformer when one computer can run two different tasks?

Will it break the bank?

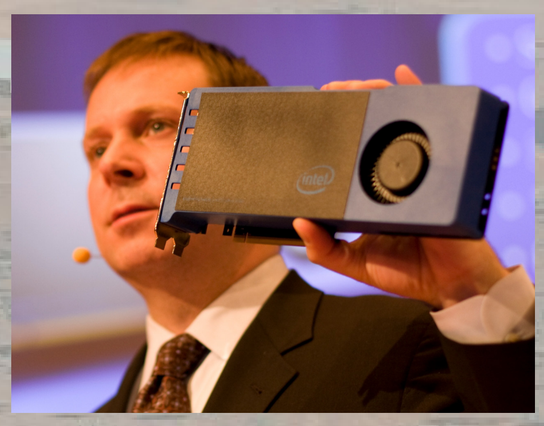
The correlation problem is sufficiently parallel that highly-parallel architectures can be used with great efficiency. The example scenario below uses two clusters of NVIDIA GPU cards – one for the F-step and the other for the X-step – and assumes computational power doubles every two years and bandwidth doubles every four years. In this worst case, the whole system (dishes and aperture arrays forming 480 beams) costs only €3m and dissipates 500kW. A time-division correlator may be achievable at half the cost.





Candidate platforms

An NVIDIA C2070 Tesla card (2009), offering a peak performance of 1030 GFLOP/s in single precision.



Intel's *Knight's Corner*, a 50-core x86 processor, slated to be commercially available in late 2012.

… and others!

Email your comments to:
D.Ford@mrao.cam.ac.uk

Swinburne Astronomy Productions