Massive Cross-correlation in Radio Astronomy with Graphics Processing Units

Research team: M.A. Clark (Harvard, NVIDIA), P.C. La Plante (Loyola University Maryland, Carnegie Melon), L.J. Greenhill (Harvard), J. Kocz (Harvard)

Cosmology: When did the first stars form, and were they like stars today? Did massive black holes arise at the same time, or were they byproducts of more gradual processes? The Universe today is shaped by these first generations of objects. Their formation is widely hypothesized to have begun about 100 million years after the Big Bang (over 13 billion years ago), but no data are yet available to test this hypothesis. This is an unexplored frontier in observational cosmology.

Radio astronomy promises to enable study of the Universe during this era via characterization of the intergalactic medium that lay between the stars and black holes for the first time. Today, this is largely hot plasma, but in the early Universe it was a vast reservoir of cold neutral Hydrogen gas. A fraction of this fed the formation of compact objects under the action of gravity while the rest was ionized by ultraviolet radiation.

Pursuit of signatures that betray the formation of stars and black holes in the early Universe is limited by signal processing capacity [1,2]. The same is increasingly true within radio astronomy broadly.

Signal Processing: Telescope collecting area is a critical figure of merit, but it is not practical to build ever-larger monolithic antennas. It is possible, however, to synthesize (virtually) such an instrument by combining the signals from a large number of smaller aperture antennas via cross correlation of their received signals. The two dimensions that need to be considered in correlation are the number of antennas or signal inputs (N), and the number of frequency channels (F) required. The operation is computationally intensive due to its quadratic scaling with the number of inputs. The design presented here will scale over many orders of magnitude in both N and F.

Cross-correlation is highly parallelizable and maps ideally to GPU architecture. The Harvard X-Engine code [3,4] attains sustained 79% utilization of single precision floating-point resources on the Fermi architecture (GTX 480/580 and C2050/M2090). In previous applications of GPUs to cross correlation, only 10-30% was obtained [5, 6, 7, 8].

Innovation: Memory management and thread mapping techniques reduce bottlenecks in data transfer from device memory and in processing. Approaches include a multi-level data tiling strategy in memory and use of a pipelined algorithm with simultaneous computation and transfer of data from host to device memory. General purpose network switching provides routing of digitized (typically 8 bit) and packetized antenna signals to GPU hosts. The Harvard X-engine loads data into the GPU via the texture unit, and data are converted to the 32-bit floating-point format without taxing the host's CPU or GPU floating-point hardware. We found that for cases of 16 inputs or greater, the data transfer times on the PCIe bus are effectively hidden by computation time.

For any implementation of an algorithm, arithmetic intensity (AI) dictates maximum performance. AI is the number of floating-point operations performed per byte of information transferred. Each correlation (i.e., from a single pair of inputs for one frequency channel) is the outer-product of two complex-valued vectors of length two; the result of which is summed in an accumulator. There are $O(N^2F)$ operations. Assuming 32-bit floating-point data, each correlation requires 32 bytes of input and 64 bytes of output, but only 32 flops. The AI is low: 32/96 = 1/3. However, if instead an m x n "tile" of correlation pairs is considered (see figure 1), there is significant reuse of input data among calculations of tile elements, and input memory requirements can be satisfied. The output memory traffic can also be reduced by a factor of *R* if the accumulated result is not stored until the calculations for all *R* samples are complete.

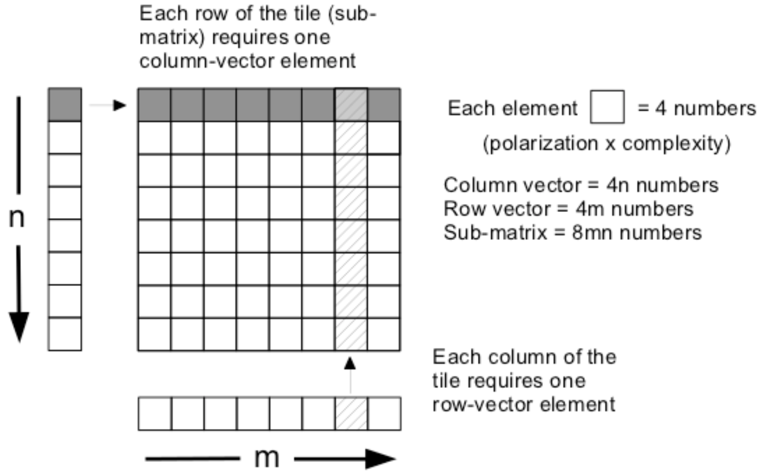
Each row of the tile (sub-matrix) requires one column-vector element
Each element □ = 4 numbers
(polarization x complexity)
Column vector = 4n numbers
Row vector = 4m numbers
Sub-matrix = 8mn numbers
n
Each column of the tile requires one row-vector element
m

Figure 1: Use of an *n* x *m* tile in memory boosts arithmetic intensity [7]. The example shown here refers to complex input data with two inputs per antenna, corresponding to the orthogonal polarizations of radiation incident.

At large *R*, the output memory traffic becomes negligible, and the arithmetic intensity can be made arbitrarily large by increasing the tile size, however, the number of registers imposes a practical limit on the size of the tile to 2x2 for Fermi systems. We overcome the processing limitations created due to this by using a multi-level memory tiling strategy, where the matrix is filled only at this smallest tile size, and other levels are used to store the input vectors only. Coalescing of memory access is necessary for high performance and demonstrated to require explicit software management of cache.

Figure 2 shows the performance of the X-engine kernel for a GTX 580 as a function of the number of pairs of inputs (N total) and frequency channels (F), with a fixed integration R=1024. Tiling is also applied to mapping of thread blocks onto the correlation matrix. The matrix is Hermitian, so only the lower triangular part is computed, with threads assigned to the respective diagonal and sub-diagonal elements. While peak floating point utilization is 79% of theoretical (single precision) for Fermi, if background instructions are included in the tally (i.e., general overhead such as loading registers) the kernel actually operates at 91% of peak. However, this depends on the dominance of the $O(N^2)$ computation over the O(N) transfer of data. For small numbers of inputs, the performance is reduced as well because thread blocks are more coarsely distributed (stair-stepwise) along the diagonal and more are wasted (i.e., above the diagonal). However, increasing the number of frequency channels improves utilization and thus performance.

Conclusions: Cross correlation in radio astronomy scales as $O(N^2F)$, and optimization is critical for arrays of hundreds and thousands of antennas, such as those designed to characterize the first generations of stars and black holes, which set the evolutionary course of the Universe. We achieve a peak computing speed of > 1 TFlop/s on a single Fermi GPU, with a sustained 79% utilization.

Radio telescopes are normally constructed in remote areas, and perennial concerns are power consumption and density. Quadratic scaling in correlation focuses attention on GPUs in this regard. As operations count per Watt increases, superior speed in code development and computational flexibility for GPUs may be anticipated to figure more prominently in full-cost accounting relative to ASICs and FPGAs.

Moreover, a greatly shortened cycle of correlator development and deployment can be critical to achieve the best science return. ASIC-based correlators requiring development over substantial fractions of a decade have been deployed for relatively small arrays (N < 100). GPU-based instruments with just a year or two of development will at least match these with rates up to about 100 TFlop/s, such as the Large Aperture Experiment to Detect the Dark Ages [1]. Beyond this, instruments requiring *O*(10) PFlop/s are anticipated in this decade, and full deployment of the Square Kilometer Array (SKA) thereafter will drive requirements into the Exascale regime. The optimizations applied in the Harvard X-engine code and resulting high efficiency advance the feasibility of GPU implementations for the massive radio astronomical correlators on the horizon.

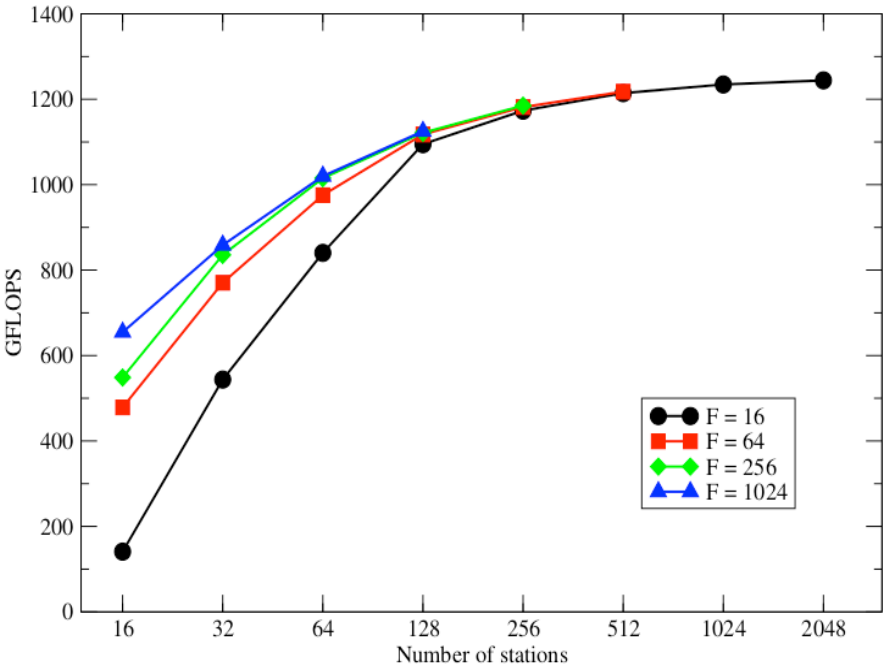

Figure 2: GTX 580 performance as a function of the number of stations (i.e., pairs of N inputs), and number of frequency channels (F) for a fixed integration length R=1024. As the number of stations increases, more threads are used and with less waste, resulting in a greater efficiency. An increase in F also boosts performance, but N^2F is limited by available memory (3 GB here). Adapted from [3].

[1] Greenhill, L. and Bernardi, G. (2012). HI Epoch of Reionization Arrays. In S. Komonjinda, Y. Kovalev, and D. Ruffolo (Eds.), 11th Asian-Pacific Regional IAU Meeting 2011, Volume 1 of NARIT Conference Series.

[2] Lonsdale, C.J., et al. (2009). The Murchison Widefield Array: Design Overview. Proc. IEEE 97, 1497–1506.

[3] Clark, M.A., La Plante, P.C., and Greenhill, L.J. (2012). Accelerating Radio Astronomy Cross-Correlation with Graphics Processing Units. IJHPCA, in press.

[4] https://github.com/GPU-correlators/xGPU

[5] Harris, C., Haines, K., and Staveley-Smith, L. (2008). GPU accelerated radio astronomy signal convolution. Exp. Astron. 22, 129–141.

[6] Wayth, R.B., Greenhill, L.J., and Briggs, F.H. (2009). A GPU- based Real-time Software Correlation System for the Murchison Widefield Array Prototype. PASP 121, 857–865.

[7] Williams, S.W., Waterman, A., and Patterson, D.A. (2008). Roofline: An Insightful Visual Performance Model for Floating-Point Programs and Multicore Architectures. Technical Report UCB:/EECS-2008- 134, EECS Department, University of California, Berkeley.

[8] Schaaf, K. and Overeem, R. (2004). COTS Correlator Platform. Exp. Astron. 17 (1-3), 287–297.