A GPU based solution for distributed FX correlation

Richard Dodson, ICRAR
Dick Ferris, CSIRO
Chris Phillips, CSIRO
VLBI Correlation (traditional)

Antenna

- Fringe rotate (and delay)
- Correlate

High Bandwidth Analogue signal
Limited bandwidth
Limited bits/samp (~12%)

ADC
2 bits

Limited bandwidth
Limited bits/samp (~12%)

Channelisation

Fringe Rotation (~9%)

Cross Correlation

Multiple Antennae

Total losses: 36%

The talk is not about adding GPUs to the DFX software correlator!!
Correlator Losses

- Conversion of analog to digital
  2 bit conversion (4 level) -12%

- Delay and rate correction
  Fringe rotation (3 bit 2 level) -6%

- Discrete Delay Step
  Delay for Centre of band Δν/2 -3%

**TOTAL**

-22%
## Correlator Losses

- **Conversion of analog to digital**
  - 2 bit conversion (4 level) \( \text{TOTA} \)
  - More Worse than -12%

- **Delay and rate correction**
  - Fringe rotation (3 bit 2 level) -6%

- **Discrete Delay Step**
  - Delay for Centre of band \( \Delta v/2 \) -3%

**Total**

*Much Worse than* -22%
VLBI Correlation (alternative)

- Antenna
- High Bandwidth Analogue signal
- Limited bandwidth
- Channelisation
- Fringe Rotation
- Limited bits/samp (~12%)
- 2 bit
- Cross Correlation
- Total losses: 22%

The correlator is distributed across the VLBI array.

- Record (or transmit)
- Load into correlator

13/08/09
Gains from pre-encoding

- Fringe rotation and channelisation at high bits per sample: Near zero losses.
- RFI Handling: Channel excision at high bit levels
- Optimum 2-bit encode: Better compression.
- Transfer of processing power to antenna hardware: Better use of processing power.
- Reduction in correlator hardware requirements: Support >GBps without upgrading correlator hardware.
GPUs - massively parallel processing

- Graphical Processing Units are designed to process the screen pixels, in parallel
- The same processing power is now being turned to `similar’ tasks
- New languages (Cuda & OpenCL) are hiding the Graphics Primitives
- Works well for suitable tasks
GPU - massively parallel processing

GPUs Are Getting Faster, Faster

© NVIDIA Corporation 2008
GPUs - massively parallel processing

GPUs Are Getting Faster

Figures from NVIDIA GPUs – massively parallel processing

Grid

Block (0, 0)

Shared Memory

Registers

Thread (0, 0)

Local Memory

Global Memory

Constant Memory

Texture Memory

Block (1, 0)

Shared Memory

Registers

Thread (1, 0)

Local Memory

Global Memory

Constant Memory

Texture Memory

© NVIDIA Corporation 2008
GPUs - limitations

- GPUs sit on the PCI bus
  - The data I/O bandwidth is limited
- GPUs have (had) limited processing power
  - Previously no floats, doubles just now available
- Memory access needs careful handling
  - Memory between different threads (i.e. pixels) has slow access
LBA plans towards GBps VLBI

- For ATCA it will be CABB based up to 2GHz (10bit)
- PKS/Mopra will use the DFB3 1GHz (8bit)
- Aiming to achieve 8-16 Gbps
- Correlation on DiFX or CABB

Chris will speak on this after lunch
GPU plans towards GBps VLBI

- Develop a auto-correlator (needed) for the current system
  
  Hardware must sit within current LBADR

- Develop fringe rotation & encoding on-card

- Replace integration-on-card with streaming to ethernet

- Massively increase processing power for next-gen eVLBI
VLBI Correlation (alternative)

- IF chain (down-convert and filter)
- A2D conversion at 8-bit
- Channelise
- Fringe rotate
- Mitigate RFI
- Compress to 2-bit
- Record (or transmit)

On GPU
GPU steps so far

- Bare Bones System: Dual streams, transfer one channelise other. Main delay is in FFTs
- Adaption of LBA program `fauto` to `gauto`:
- Ugly, but working, single stream. Main delay is conversion of bits and memory coalescing
- Next steps: improve coding, improve coalescing, shift demangling onto card,
- New assistance, cuda-gdb now available
- GPU steps so far
  - Bare Bones System: Dual streams, transfer one channelise other. Main delay is in FFTs
  - Adaption of LBA program *auto* to *gauto*:
    - Ugly, but working, single stream. Main delay is conversion of bits and memory coalescing
- Next steps: improve coding, improve coalescing, shift demangling onto card,
- New assistance, cuda-gdb now available
GPU steps so far

- **Bare Bones System**: Dual streams, transfer one channelise other. Main delay is in FFTs

- **Adaption of LBA program**
  - fauto to gauto
  - Ugly, but working, single stream. Main delay is conversion of bits and memory coalescing

Next steps: improve coding, improve coalescing, shift demangling onto card,

New assistance, cuda-gdb now available
GPU steps in near term

- Develop our skills for Antenna based channelization: Now
- Providing a real-time auto-correlator: Now, but needs improvement
- Roll out 9880GT onto all LBA antennae: Costs small (<$2000)
- Outcomes from ARC grant to do this: November
- Outcomes from NVIDIA support for the same: Soon
In conclusions

One can improve the efficiency of the VLBI correlators.
Main gain will be in achieving the ideal A2D conversion.
This can be done with GPUs.
ICRAR is developing their expertise in this area &
Converting existing software auto-correlators.
Developing the code for data encoding/decoding.
fringe-rotation.