LOFAR: A decade of correlating in a software telescope

Jan David Mol,
The 7th Int’l VLBI Technology Workshop, Krabi, Thailand, 2018
• 48 - 96 antennas, 2 pol
• 10 – 240 MHz
• 200 MHz 8-bit complex
• Beamformed on FPGA
• Baselines:
  • $\leq 2000$ km
  • 38 Dutch stations, 13 Intl
• FoV:
  • $300$ deg$^2$ @ 15 MHz
  • $1.3$ deg$^2$ @ 240 MHz
• Resolution:
  • $\leq 3.3$ arcsec @ 15 MHz
  • $\leq 0.2$ arcsec @ 240 MHz
• Clock:
  • Shared @ 24 stations
  • GPS + Rubidium
  • (White Rabbit)
  • Central phase alignment
• Network:
  • $\leq 20$ ms latency
  • 3 – 6 (9) Gbps/station
More central processing:
- Tied-array beams (Stokes I/IQUV)
- Per-station time series (“fly’s eye”)

Station (FPGA)

Antenna → ADC + Filter → Beam Former → PPF + Band Selection

1-3x per station

Correlator

Buffer + Sample Alignment

Phase Alignment

Bandpass Correction

Int->Float + PPF

Correlation + Integration

WAN

51x

Station (FPGA)

LAN

488 x 0.1 - 60s (100 Gbps)

Disk

Flagging + Averaging

Calibration

Imaging

Archive sites

Storage & Off-line processing

488 x 105 kHz (3 Gbps)
Early days: IBM BlueGene

- 2005: IBM BlueGene/L
  - 1 rack + 2 head nodes
  - 1024x PPC440 @ 700 MHz
- 2008: IBM BlueGene/P
  - 1 rack + 2 head nodes
  - 1024x PPC450 @ 850 MHz
  - Dual FP pipeline (“Double Hummer”)
    - 1 FLOP/byte
  - 1 thread/core
  - Fast interconnects
    - 3D Torus + Tree
  - I/O: 64x 10GbE
BlueGene software architecture
BlueGene parallelisation

- Output to 100 data writers with local storage
- Custom protocol for enough bw
- Point-to-point
- Avoid link overload
Now: GPU servers (“COBALT”)

- 2013: COBALT1
  - 8x Dell PowerEdge T640 (Xeon E5-2660)
  - 16x NVIDIA Tesla K10
  - 16x FDR Infiniband (56 Gbps)
  - I/O: 32x 10GbE
- Dec 2018: COBALT2
  - 1x Dell PowerEdge R440 (head node)
  - 11x Dell PowerEdge R740 (Xeon Gold 6140)
  - 22x NVIDIA Tesla V100
  - 23x EDR InfiniBand (100 Gbps)
  - I/O: 80x 10GbE
- COTS
- Balanced PCIe layout
COBALT software architecture
COBALT parallelisation

- Demands tuning
- Not thread safe!
- Global Lock + MPI_I*
- No work outside MPI calls
- Polling thread
- Minimise #transfers

#omp parallel sections + thread-safe queues

Driver API + JIT compilation

COBALT2: mounted locally
Compute power

### Compute power

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TFLOPS (single precision)</td>
<td>6</td>
<td>14</td>
<td>73</td>
<td>308</td>
</tr>
</tbody>
</table>

### Unit count

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Units</td>
<td>1.024</td>
<td>1.024</td>
<td>32</td>
<td>22</td>
</tr>
</tbody>
</table>

### Power consumption

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>KW</td>
<td>22</td>
<td>22</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

### Compute core count

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cores</td>
<td>2.048</td>
<td>4.096</td>
<td>49.152</td>
<td>112.640</td>
</tr>
</tbody>
</table>
Compute bandwidth

LOFAR input bandwidth

Interconnect bandwidth

Memory

Memory bandwidth
COBALT2: “LOFAR Mega Mode”

<table>
<thead>
<tr>
<th>Science (simultaneous)</th>
<th>Processing Type</th>
<th>#P100s</th>
<th>#V100s (est)</th>
</tr>
</thead>
<tbody>
<tr>
<td>All-sky imaging survey</td>
<td>Correlation</td>
<td>2.8</td>
<td>1.9</td>
</tr>
<tr>
<td>Targets of opportunity</td>
<td>Correlation</td>
<td>0.3</td>
<td>0.2</td>
</tr>
<tr>
<td>Pulsar &amp; Fast transients survey</td>
<td>Tied-array beam forming</td>
<td>4.0</td>
<td>2.6</td>
</tr>
<tr>
<td>Recombination lines</td>
<td>Tied-array beam forming</td>
<td>1.6</td>
<td>1.1</td>
</tr>
<tr>
<td>Pulsar timing &amp; RM monitoring</td>
<td>Tied-array beam forming</td>
<td>4.9</td>
<td>3.3</td>
</tr>
<tr>
<td>Ionospheric scintillation</td>
<td>Per-station time series</td>
<td>2.7</td>
<td>1.8</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>16.3</strong></td>
<td><strong>10.9 (of 22)</strong></td>
</tr>
</tbody>
</table>

- Highly different settings/submode
- Compute power to spare! Could do f.e.:
  - More tied-array beams
  - Better signal quality (on-line RFI filtering, Doppler tracking, dedispersion)
  - Real-time (uncalibrated) imaging
COBALT challenges

• Drop-in replacement
• Hardware/software co-design
  • Real-time processing on non-real-time systems
• EU Tendering (COBALT2)
  • Squeezes vendors
  • Cannot demand end-to-end performance
  • No consumer-grade GPUs, no 32GB V100s...
• Port all code PPC → OpenCL → CUDA
  • Wide range in problem dimensionality
  • JIT kernel compilation
• Driver incompatibility
  • Shared memory vs pinned memory clashes
COBALT1 Temperature Control

GPUs 70C @ full load

GPUs 75C @ idle

GPUs 50C @ full load
### Fitness for purpose?

<table>
<thead>
<tr>
<th>Property</th>
<th>BlueGene</th>
<th>COBALT (GPU servers)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-time processing</td>
<td>✅ Little OS interference</td>
<td>✅ Overdimension CPU/DRAM</td>
</tr>
<tr>
<td>Stability</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>Platform</td>
<td>✅ Integrated</td>
<td>✗ Frankenstein</td>
</tr>
<tr>
<td>Workload distribution</td>
<td>✗ 10-20 compute units/unit of work</td>
<td>✅ 15-20 units of work/compute unit</td>
</tr>
<tr>
<td></td>
<td>✗ Avoid clashes on tree/torus</td>
<td>✅ Star network</td>
</tr>
<tr>
<td></td>
<td>✗ Custom low-level tree protocol</td>
<td>✅ Tune common software</td>
</tr>
<tr>
<td>Staging memory</td>
<td>✗ Little</td>
<td>✅ Much</td>
</tr>
</tbody>
</table>
## Programmability

<table>
<thead>
<tr>
<th>Property</th>
<th>BlueGene</th>
<th>COBALT (GPU servers)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute kernels</td>
<td>X Assembly (to use special FP instructions)</td>
<td>✓ Common GPU techniques</td>
</tr>
<tr>
<td>New outputs</td>
<td>X Manual routing...</td>
<td>✓ Local disk write (Lustre)</td>
</tr>
<tr>
<td>Debugging</td>
<td>X Track work across 4096 cores</td>
<td>✓ Few units, easy work distribution</td>
</tr>
<tr>
<td></td>
<td>X Remote debugging</td>
<td>✓ Common GPU tools</td>
</tr>
<tr>
<td>Crash recovery</td>
<td>X 45-minutes (not uncommon)</td>
<td>✓/X 15-minutes (rare)</td>
</tr>
<tr>
<td>Expertise</td>
<td>X Rare (PPC assembly)</td>
<td>✓ More common</td>
</tr>
</tbody>
</table>
Conclusions

• Yes, GPUs are powerful and hot
  • We know that
• Whole COTS package makes it shine:
  • Work vs compute unit ratio (easy work distribution),
  • Network topology, global fs (easy input/output distribution),
  • Powerful middleware (many problems are solved for you)
• Allows focus on flexibility
• Allows more science
Thank you