Acceleration of parallel algorithms using a heterogeneous computing system

Ra Inta,
Centre for Gravitational Physics,
The Australian National University

NIMS-SNU-Sogang Joint Workshop on Stellar Dynamics and Gravitational-wave Astrophysics, December 18, 2012
Overview

1. Why a new computing architecture?
2. Hardware accelerators
3. The Chimera project
4. Performance
5. Analysis via Berkeley’s ‘Thirteen Dwaves’
6. Issues and future work
7. Conclusion
Why do we need new computing architectures?

- We live in an increasingly data-driven age
- LHC experiment alone: data rate of 1 PB/s
- Many data-hungry projects coming on-line
- Computationally limited (most LIGO/Virgo CW searches have computation bounds)
Demand side…

Performance trend of global top 500 supercomputing systems

(adapted from www.top500.com, August 2012)
…supply side

- Intel discontinued 4GHz chip
- Speed walls
- Power walls
- Moore’s Law still holds, but for multicore

Each Year We Get Faster More Processors

Historically:
Boost single-stream performance via more complex chips.
Now:
Deliver more cores per chip (+ GPU, NIC, SoC).
The free lunch is over for today’s sequential apps and many concurrent apps. We need killer apps with lots of latent parallelism.
Time for a change in high performance computing?

• Focus on throughput, rather than raw flop/s (e.g. Condor on LSC clusters)
• Focus on power efficiency (top500 ‘Green list’)

Hardware acceleration

• Response to computational demand
• Two most common types: Graphical Processing Unit (GPU) and Field Programmable Gate Array (FPGA)
The Graphical Processor Unit

- Subsidised by gamers
- Most common accelerator
- Excellent support (nVidia)
The Field Programmable Gate Array

Programmable hardware (‘gateware’) replacement for application-specific integrated circuits (ASICs)
Cultural differences

- GPUs: software developers
- FPGAs: hardware, electronic engineers

FPGA development is a lot less like traditional microprocessor programming! (VHDL, Verilog)
<table>
<thead>
<tr>
<th>Platform</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Analysis ‘workhorse,’ multi-tasking</td>
<td>Power hungry, limited processing cores</td>
</tr>
<tr>
<td>GPU</td>
<td>Highly parallel, fairly simple interface (e.g. C for CUDA)</td>
<td>Highly rigid instruction set (don’t handle complex pipelines)</td>
</tr>
<tr>
<td>FPGA</td>
<td>Unrivalled flexibility and pipelining</td>
<td>Expensive outlay, specialised programming interface, prohibitive development time</td>
</tr>
</tbody>
</table>
Choosing the right hardware

Highly algorithm dependent, e.g.

- Dense linear algebra: GPU
- Logic-intensive operations: FPGA
- Pipelining: FPGA
The ‘Chimera’ project

• Appropriately(?) combine benefits of three hardware classes
• Commercial, off the shelf (COTS) components
• Focus on high throughput, energy efficiency

(Image: E. Koehn)
Initial design

COTS constraints
<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Vendor</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel</td>
<td>i7 Hexacore</td>
</tr>
<tr>
<td>GPGPU</td>
<td>nVidia</td>
<td>Tesla C2075</td>
</tr>
<tr>
<td>FPGA</td>
<td>Altera</td>
<td>Stratix-IV</td>
</tr>
</tbody>
</table>
Photograph of an implementation of the system. Because of the three hardware classes, we named the system the ‘Chimera,’ after the mythical Greek beast with three different heads.
Performance

Monte Carlo calculation of $\pi$

\[ A_T = 4R^2 \]
\[ \frac{A_C}{A_T} = \frac{N_C}{N_T} = \pi/4 \]

1: Create $N_T$ uniformly distributed random positions within unit square.
2: Count number of points $N_C$ satisfying $x^2 + y^2 < 1$
3: Calculate $4N_C/N_T$ and display

Excellent at generating random numbers
Excellent at dense linear algebra

FPGA

CPU

GPU
Throughput:

- FPGA alone: $24 \times 10^9$ samp/s
- GPU alone: $2.1 \times 10^9$ samp/s
- CPU alone: $10 \times 10^6$ samp/s

Surprising!

- FPGA pipeline was highly optimised
- Expect GPU to perform better with more complicated (floating point) integration problems
- Expect CPU to have even worse performance
Normalised cross-correlation

$$\sum\sum_{X,Y} A[x, y] \cdot B[x + X, y + Y]$$

- image processing
- synthetic aperture arrays (SKA, VLBA)
- video compression
Searched 1024×768 pixel image (16-bit greyscale) for a 8×8 template

- Numerator
  
  GPU, 158 frame/s

- Denominator
  
  GPU: 894 frame/s
  FPGA: 12,500 frame/s

So much for a Graphical Processor Unit!

Best combination: GPU + FPGA
• Data analysis
• Interface GPUs with experimental FPGA DAQ systems (GMT, quantum cryptography)

The Thirteen Dwarves of Berkeley

- Phillip Colella identified 7 classes of parallel algorithms (2004)
- A dwarf: “an algorithmic method encapsulating a pattern of computation and/or communication”
- UC Berkeley CS Dept. increased to 13

<table>
<thead>
<tr>
<th>Dwarf</th>
<th>Examples/Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense Matrix</td>
<td>Linear algebra (dense matrices)</td>
</tr>
<tr>
<td>Sparse Matrix</td>
<td>Linear algebra (sparse matrices)</td>
</tr>
<tr>
<td>Spectral</td>
<td>FFT-based methods</td>
</tr>
<tr>
<td>N-Body</td>
<td>Particle-particle interactions</td>
</tr>
<tr>
<td>Structured Grid</td>
<td>Fluid dynamics, meteorology</td>
</tr>
<tr>
<td>Unstructured Grid</td>
<td>Adaptive mesh FEM</td>
</tr>
<tr>
<td>MapReduce</td>
<td>Monte Carlo integration</td>
</tr>
<tr>
<td>Combinational Logic</td>
<td>Logic gates (e.g. Toffoli gates)</td>
</tr>
<tr>
<td>Graph traversal</td>
<td>Searching, selection</td>
</tr>
<tr>
<td>Dynamic Programming</td>
<td>‘Tower of Hanoi’ problem</td>
</tr>
<tr>
<td>Backtrack/Branch-and-Bound</td>
<td>Global optimization</td>
</tr>
<tr>
<td>Graphical Models</td>
<td>Probabilistic networks</td>
</tr>
<tr>
<td>Finite State Machine</td>
<td>TTL counter</td>
</tr>
</tbody>
</table>
*Fixed point
\(^\wedge\)Floating point
Issues/future work

- Kernel module development
- Scalability is problem-size dependent
- FPGA development is labour-intensive
- Pipeline development
- GW data analysis!
Issues/future work
Issues/future work

• Kernel module development
• Scalability is problem-size dependent
• FPGA development is labour-intensive
• Pipeline development
• GW data analysis!
Conclusion

• We are in a transition period for HPC architecture
• Can exploit highly heterogeneous, COTS, hardware accelerators for certain algorithms
• Powerful and energy efficient
• Requires further development

More information:
Thanks to NIMS and Seoul National University!

- Dr Sang Hoon Oh
- Dr John J. Oh
Thanks for listening!

Ra.Inta@anu.edu.au