Emerging HPC technologies: back to the future?

Stan Scott
School of Electronics, Electrical Engineering and Computer Science
The Queen's University of Belfast, Belfast, BT7 1NN, UK
The throes of a revolution?

Games arena
• NVIDIA GPU
• STI Cell processor

HPC arena
• Clearspeed floating-point accelerator

Embedded systems arena
• FPGA

Heterogeneous chips
• Intel Larrabee

Heterogeneous systems
• IBM Roadrunner

<table>
<thead>
<tr>
<th></th>
<th>Single GF</th>
<th>Double GF</th>
<th>Bandwidth GB/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Core2Quad Extreme</td>
<td>192</td>
<td>96</td>
<td>12.6</td>
</tr>
<tr>
<td>NVIDIA GTX 2800</td>
<td>930</td>
<td>93</td>
<td>130</td>
</tr>
</tbody>
</table>

Myth or Reality
1. Rebirth of SIMD?

When vector processing declined in favour of computational clusters, SIMD processing appeared to fade from the mainstream. It could be argued that SIMD processing is now undergoing a “rebirth” as it is becoming a centrepiece in modern architectures such as the Cell BE [2], modern general purpose graphics processors [16] and likely the fruits of the Intel Larrabee project [3] as well.

Massive thread-level parallelism inherent on each SCF iteration


Transference of bones from one graveyard to another?
2. Non compliance with IEEE 754 – does it matter?

“The major current challenge in running codes of this type on the GPU arises from the lack of fully compliant IEEE floating point implementations.”

```
S = x[1];
C = 0;
for(j=2; j<=n; j++){
  Y = x[j]-C;
  T = S + Y;
  C = (T-S)-Y;
  S = T;
}
```

Fig. 5. KSF corrects for rounding error in matrix multiplication. The resultant matrix is $1000 \times 1000$, and the operand data is sampled from a uniform distribution $[0, 1]$.

3. Single vs double precision fp performance

“On modern architectures, the performance of 32-bit operations is often at least twice as fast as the performance of 64-bit operations. By using a combination of 32-bit and 64-bit floating point arithmetic ... performance ... can be significantly enhanced while maintaining the 64-bit accuracy of the resulting solution.”

Iterative refinement for dense systems, \( Ax = b \), can work this way.

\[
\begin{align*}
L U &= \text{lu}(A) & \text{SINGLE} & O(n^3) \\
x &= L\backslash(U\backslash b) & \text{SINGLE} & O(n^2) \\
r &= b - Ax & \text{DOUBLE} & O(n^2) \\
\text{WHILE} & \| r \| \text{ not small enough} \\
z &= L\backslash(U\backslash r) & \text{SINGLE} & O(n^2) \\
x &= x + z & \text{DOUBLE} & O(n^1) \\
r &= b - Ax & \text{DOUBLE} & O(n^2) \\
\end{align*}
\]

END

- Requires extra storage, total is 1.5 times normal;
- \( O(n^3) \) work is done in lower precision
- \( O(n^2) \) work is done in high precision

3. Single vs double precision fp performance

![Graphs showing performance comparison between single and double precision methods for LU and Cholesky solves on a Cell Broadband Engine.](image)

*Fig. 1. Mixed precision, iterative refinement method for the solution of dense linear systems on the STI Cell BE processor.*

4. Memory wall

Science News

More Chip Cores Can Mean Slower Supercomputing, Simulation Shows

ScienceDaily (Jan. 15, 2009) — The worldwide attempt to increase the speed of supercomputers merely by increasing the number of processor cores on individual chips unexpectedly worsens performance for many complex applications, Sandia simulations have found.

A Sandia team simulated key algorithms for deriving knowledge from large data sets. The simulations show a significant increase in speed going from two to four multicores, but an insignificant increase from four to eight multicores. Exceeding eight multicores causes a decrease in speed. Sixteen multicores perform barely as well as two, and after that, a steep decline is registered as more cores are added.

The multicore dilemma: more cores on a single chip don’t necessarily mean faster clock speeds, a Sandia simulation has determined. (Credit: Photo by Randy Montoya)

Reference
- Blue Gene
- MRAM
- Computer software
- Supercomputer
- set of wires used to carry memory addresses and data to and from the system RAM

Distributed Computing
Server sprawl is a problem. Control it with a virtual network from BMC.
www.BMC.com

Dual Core Processors
Increase Business Performance & Reduce Your
4. Memory wall

SOFTWARE/ALGORITHMS FOLLOW HARDWARE EVOLUTION IN TIME

<table>
<thead>
<tr>
<th>LINPACK (70’s) (Vector operations)</th>
<th>Rely on - Level-1 BLAS operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAPACK (80’s) (Blocking, cache friendly)</td>
<td>Rely on - Level-3 BLAS operations</td>
</tr>
<tr>
<td>ScaLAPACK (90’s) (Distributed Memory)</td>
<td>Rely on - PBLAS Message Passing</td>
</tr>
<tr>
<td>PLASMA (00’s) New Algorithms (many-core friendly)</td>
<td>Rely on - a DAG-based scheduler - a block data layout - some extra kernels</td>
</tr>
</tbody>
</table>

PLASMA

Novel data formats like Block Data Layout (BDL) improve locality of reference to memory and higher reuse of data in memories that are closer to cores (caches or scratchpad memories).

http://icl.cs.utk.edu/plasma
FPGAs

Allow us to build a custom CPU that only has the instructions that we need to match our algorithm.

- Profile to identify code to run on FPGA
- Modify code to use FPGA C language (such as Handel-C, Mitrion-C, etc.)
- Compile this into a hardware description language (VHLD or Verilog)
- Perform FPGA place-and-route and product FPGA “bitfile”
- Download bitfile to FPGA
- Compile complete application and run on host processor and FPGA

Six Xilinx Virtex-4 Field Programmable Gate Arrays (FPGAs) per chassis

HPC/NA Workshop, 25/26 January 2009

Emerging HPC technologies: back to the future?
QP: A Heterogeneous Multi-Accelerator Cluster

QP: Quadro Plex - applications

<table>
<thead>
<tr>
<th>Application</th>
<th># of CPU cores used</th>
<th># of GPUs used</th>
<th># of FPGAs used</th>
<th>non-accelerated application performance</th>
<th>accelerated application performance</th>
<th>overall speedup achieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAMD (molecular dynamics) [20]</td>
<td>60</td>
<td>60</td>
<td></td>
<td>0.471 sec/step</td>
<td>0.085 sec/step</td>
<td>5.5X</td>
</tr>
<tr>
<td>WRF (weather prediction) [21]</td>
<td>48</td>
<td>48</td>
<td></td>
<td>79.9 sec</td>
<td>64.9 sec</td>
<td>1.23X</td>
</tr>
<tr>
<td>ACF (cosmology) [22]</td>
<td>8</td>
<td>8</td>
<td></td>
<td>5,537.7 sec</td>
<td>880.8 sec</td>
<td>6.3X</td>
</tr>
<tr>
<td>ACF (cosmology) [22]</td>
<td>32</td>
<td>32</td>
<td></td>
<td>1,476.8 sec</td>
<td>70.1 sec</td>
<td>21.1X</td>
</tr>
</tbody>
</table>

Table 2. Applications developed to run on the multi-core/GPU/FPGA cluster

Tower of Babel – “.. Come let us go down and confuse their language so that they will not understand each other ..”, Genesis 11
HONEI: A collection of libraries for numerical computations targeting multiple processor architectures.

**Level of Abstraction**
- High Level Operations & Solvers
- HONEI Linear Algebra Functions & HONEI Math & SWE Kernels
- HONEI Utility Functions & HONEI Containers
- HONEI Architecture Based Backends
  - Cell BE
  - x86 / SSE
  - GPU
  - ..

**Figure 1:** HONEI bottom up structure and components, from the underlying hardware to the application level.

**Frontend**
- Linear Algebra Library
  - dense, banded and general sparse matrices
  - rich subset of BLAS level 1, 2, 3
- Math Kernel
  - iterative linear system solvers
  - quadrature
  - interpolation

**Backend**
- SSE2
  - libhoneibackendssse
- CUDA
  - libhoneibackendscuda
  - transparent memory scheduler
- Cell
  - libhoneibackendscell

---


HPC/NA Workshop, 25/26 January 2009  Emerging HPC technologies: back to the future?
HONEI: A collection of libraries for numerical computations targeting multiple processor architectures.

Climbing a mountain to reach the moon?

The basic idea is to execute a simple defect correction scheme in double precision on the CPU, which is preconditioned by a multigrid solver on the GPU. After two full multigrid iterations, the result is transferred back to the CPU to advance the solution.

Sequoia is a programming language that is designed to facilitate the development of memory hierarchy aware parallel programs that remain portable across modern machines with different memory hierarchy configurations.  
http://www.stanford.edu/group/sequoia/
Challenging times ahead for HPC

- Algorithms that can exploit the memory hierarchy
- Mixed precision algorithms
- Performance portability
- Checkpointing algorithms
- Adaptive and self tuning algorithms
- New ways of programming
“... As a nation we are poorly equipped to address these challenges. There is a lack of cohesion across the disciplinary groups that need to be brought together, namely mathematics, computer science, engineering and domain scientists. There are few existing channels to allow a flow of knowledge and expertise across the academia/industry divide. We have an inadequately trained next generation of researchers and we do not have the required skilled workforce....”
Gather a critical mass of interdisciplinary expertise.

Develop accelerated implementations of `hot spot' solutions with performance portability.

Develop techniques to allow the maintenance of a single source tree for each code.

Development of fault tolerant algorithms through checkpointing.

Characterize computation, communication and I/O patterns and abstract general principles and guidelines for the design of applications and algorithms for emerging and future HPC systems.

17 HPC researchers including Stan Scott, George Constantinindes, Peter Knowles, David Walker, Fred Manby, Mike Ashworth, Neil Sandham et al.
Many-core and Reconfigurable Supercomputing Network

The Many-core and Reconfigurable Supercomputing Network (MRSN) is an EPSRC-funded initiative by Imperial College, University of Manchester, University of Oxford and Queen's University Belfast to promote the use of FPGAs, many-core GPUs and other accelerators for scientific computation in a range of application areas such as healthcare, digital media and transportation, all of which are of importance within EPSRC’s Digital Economy programme.

The goals are to help each other with these exciting new technologies, build bridges between application scientists and computing experts, maintain/develop links with international experts, and prepare joint proposals for future research funding.

Mailing list
People can join the MRSN mailing list and/or look at the mail archives here.

Contact:
Mike.Giles@maths.ox.ac.uk

http://www.oerc.ox.ac.uk/research/many-core-and-reconfigurable-supercomputing

Many-Core and Reconfigurable Supercomputing Conference
March 25th - 26th, 2009

Zuse Institute Berlin, Germany

- Algorithms and programming models
- High-level programming environments, languages and tools
- Applications on hardware accelerators
- Performance modelling and prediction
- Communication in multi-accelerators platforms
- Scaling to use thousands of many-core or reconfigurable processors
- System environments and scheduling
- Architectures and system evaluation
- Standard proposals

Paper submissions due: January 25, 2009
## Emerging HPC technologies

### Table 3: Theoretical vs. measured Gflop/s rates on 32 and 64-bit matrix-matrix multiplication

<table>
<thead>
<tr>
<th>Device/Platform</th>
<th>Theoretical SGEMM</th>
<th>Measured SGEMM</th>
<th>Theoretical DGE MM</th>
<th>Measured DGE MM</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA GTX 280 (GPU^a ATI FireStream)</td>
<td>250</td>
<td>60(^f)</td>
<td>not possible</td>
<td>not possible</td>
</tr>
<tr>
<td>FPGA(^b) Virtex-4 V4LX200(^d)</td>
<td>40(^c)</td>
<td>4(^d)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ClearSpeed</td>
<td>96</td>
<td>96</td>
<td>45(^f)</td>
<td></td>
</tr>
<tr>
<td>Cell(^e)</td>
<td>205</td>
<td>201</td>
<td>15</td>
<td>12</td>
</tr>
<tr>
<td>Xeon 3.8 GHz (1 core)</td>
<td>15</td>
<td>11(^c)</td>
<td>8</td>
<td>6(^d)</td>
</tr>
<tr>
<td>Woodcrest 2.6 / GHz (1 core)</td>
<td>21</td>
<td>14(^f)</td>
<td>11</td>
<td>8(^e)</td>
</tr>
</tbody>
</table>

Notes:

- a – 1 M Black-Scholes iterations run 10x typical CPU rates
- b – On some Virtex-4 chips four-bit complex dot products have a peak that corresponds to 480 Gflop/s
- c – Results from Xilinx presentation
- d – DGEMM without transfer rates has been measured at 8 Gflop/s by Scott Campbell at University of Colorado
- e – All results are from IBM paper “Cell Broadband Engine Architecture and its first implementation”
- f – Measured by HP accelerator team

• Effective Use of Many-Core and Hybrid architectures
  Dynamic Data Driven Execution
  Block Data Layout
• Exploiting Mixed Precision in the Algorithms
  Single Precision is 2X faster than Double Precision
  With GP-GPUs 10x
• Self Adapting / Auto Tuning of Software
  Too hard to do by hand
• Fault Tolerant Algorithms
  With 1,000,000’s of cores things will fail
• Communication Avoiding Algorithms
  For dense computations from $O(n \ log p)$ to $O(\log p)$
  communications
  GMRES s-step compute ($x, Ax, A^2x, \ldots, A^sx$)
IBM Cell Processor

- IBM, Toshiba, Sony (similar to the chip in the PS3)
- Each Cell contains 8 APUs.
  - An APU is a self contained vector processor which acts independently from the others.
  - 4 floating point units capable of a total of 32 Gflop/s (8 Gflop/s each)
  - 204 Gflop/s peak! 32 bit floating point; 64 bit floating point at 15 Gflop/s (3.2 GHz)
  - IEEE format, but only rounds toward zero in 32 bit, overflow set to largest
NVIDIA’s Tesla T10P

- T10P chip
  - 240 cores; 1.5 GHz
  - Tpeak 1 Tflop/s - 32 bit floating point
  - Tpeak 100 Gflop/s - 64 bit floating point

- GTX 280
  - 1 – T10P; 1.3 GHz
  - Tpeak 864 Gflop/s - 32 bit floating point
  - Tpeak 86.4 Gflop/s - 64 bit floating point
In the “old days” it was: each year processors would become faster.

Today the clock speed is fixed or getting slower.

Things are still doubling every 18-24 months.

Moore’s Law reinterpreted.
  - Number of cores double every 18-24 months.

From K. Olukotun, L. Hammond, H. Sutter, and B. Smith.
Mitrion-C program transformed to a network on chip representation (Mitrion Virtual Processor (MVP)).

All statements execute at the same time.

Mitrion-C is a **data parallel language** – semantics distinct from C.

Variables do not correspond to locations in a RAM which can only be accessed sequentially. Variables are now data packets transported around the MVP in a time dependent fashion.

Lists with foreach operation generate pipelines

Vectors with foreach operation generate wide parallelism