HPC challenges for the next years

The rising of heterogeneity and its impact on simulations

CECAM Workshop
Microscopic simulations: forecasting the next two decades
Toulouse, April 24-26 2019

Luiz Angelo STEFFENEL
angelo.steffenel@univ-reims.fr
About my team

• Luiz Anglo Steffenel
  • Associate Professor, CReSTIC Laboratory
  • CASH Team (HPC, Autonomous computing, Heterogeneity)
• Our team has a long tradition on HPC
  • ROMEO supercomputing center
  • Part of MASCa
Maison de la Simulation Champagne-Ardenne

More than 15 years associating HPC and applied computing
2013 - Biggest hybrid CPU/GPU cluster in France
   270 TFlops
   151th in Top500
   5th in Green500

2018 – Biggest academic cluster in France
   1022 Tflops
   249th in Top500
   20th in Green500
Top 500 ranking over the time

We are in a "calm" period

Besides multicore, what is the biggest "innovation" since 2008?
Hybrid architectures

- Mix of CPUs and accelerators
  - GPUs (mostly NVIDIA)
  - Other accelerators (Xeon Phi)
TOP500 – Which is the impact of accelerators?

GPUs as a way to reduce overall costs and look nice at TOP500 rank

Accelerators can deliver extra FLOPS but they add an extra heterogeneity layer → harder to explore
How to extract more from the hardware?

• GPUs are **good tools**
  • Useful with specific code parts

• Some problems are **intrinsically hard**
  • Hardware evolution helps doing *faster*, but does not reduce complexity

• Better results only come with additional software development

• Extra hardware = extra complexity
Example: n-body problems

- In 30 years
  - $10^7$ hardware
  - $10^{10}$ software
- Our problem now is that hardware is much more complex
  - Software has to struggle to control it

Source: T. Quinn
The cost of Heterogeneity

- Most of our programming models are 20+ years old (MPI, OpenMP, etc.)
  - Designed for homogeneous environments
    - Node-node, CPU-CPU, CPU-memory
- Current HPC has several layers
  - GPUs
  - Cores in a CPU
  - Multi CPUs
  - Multiple layers of memory (cache, RAM, etc.)
  - Interconnections

Good GPU Programming is not trivial
So what are GPUs good for?

• As a "piece of hardware", GPUs are no more special than co-processors for i386/i486

• Early HPC developments with GPUs started by exploring their parallel processing capabilities (SIMT)
  • GROMACS ✅
  • Fluent ✅
  • OpenFOAM ✅
  • Autodock with GPU ❌

• Performance gains limited by memory and latency constraints

Hard to code (CUDA, OpenCL, ...)

CReSTIC
The revival of Neural Networks

- GPUs are well-suited for the matrix/vector math involved in machine learning
  - Especially the famous Deep Learning
  - Data is often provided as a matrix of pixels
    - Or matrices of n-dimensions called "tensors"
    - The work can be split in several parallel tasks
  - Data is kept in the GPU memory for a long time
Is AI the future of HPC?

- Once again, it's a good tool, not the answer
- AI can help us to speed up simulations
- What AI can do for us?
  - **Unveil correlations**
    - Help improve the simulation models
    - Ex: meteorological models
  - **Identify/reproduce patterns**
    - Fill the gap between simulations steps
    - Ex: molecular docking
Ex: quantum many-body problem

- Microsoft and ETH project
- Use neural networks to represent the wave function and reduce the computing complexity
- AI does not replace the simulation models, just accelerate some steps

https://science.sciencemag.org/content/355/6325/602
AI + Simulation = Synthesis Models

AI+HPC WORKFLOW FOR ENHANCEMENT MODELING
Using Simulation Data To Train AI - Fermilab NOvA

Simulation
4.7M simulated events

Simulation Apps.
FLUKA
CRY
Cosmic Ray
Genie
GEANT

DL Training
3.7M events for training
CVN-Trained for 1 week

Inference/Testing
1M events for Testing
Neutrino detection efficiency increased by 33%

DLT Frameworks
Caffe

Source: NVIDIA

AI+HPC WORKFLOW FOR MODULATION
Al-led Experiment To Converge Faster-bose Einstein Condensate

Conventional Approach

1. Slowly adjusts each parameter
2. BEC reached in 140 experiments

AI Synthesis Modeling

ML Training
1. Trained on 10 random experiments

AI-Led Convergence
2. AI-Led experiments to converge on BEC

3. BEC reached in 10-12 experiments

Source: NVIDIA
WARNING

• Artificial Intelligence is handy, but has a (hidden) cost
• Most AI models are developed for the same kind of problems
  • Ex: image recognition
• Adapting our data is an expensive task
  • And no one dares to tell you this
• Good training requires sufficient train data
  • What to do if we lack data?
  • Worst: AI is a black box
    • No explanation on the reasoning
    • Reproducibility is not a priority
Can we rely on GPUs for general computing?

- Trends for NVIDIA/AMD
  - 7nm or less, energy constraints, interconnection speed, but...

- More and more dedicated for AI
  - Ex: TPUs from Google
  - Autonomous cars (Tesla, etc)

- WARNING: all GPU development points towards mixed precision
  - Faster, acceptable precision
  - Not adapted for all problems
Even the CPUs are changing

- Arrival of ARM processors on the HPC market
- Just an European dream?
European processor Initiative (EPI)

- Sovereignty
- Independence from US components
- Cost savings

Supercomputer Cost split

- CPU: 49%
- RAM: 16%
- IC: 11%
- supercomputer: 11%
- Infra: 24%

- +70% 3rd Parties

CPU + Memory = 65% of BOM

Developing a pan-European supercomputing infrastructure
Public Members: 1 B€; EU Financial: 486 M€; Private partners 400 M€
Europe can provide most of the elements

<table>
<thead>
<tr>
<th>Own processor?</th>
<th>Interconnect</th>
<th>Scale-up system</th>
<th>HPC system</th>
<th>AI system</th>
<th>Consulting &amp; services</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPI (ARM)</td>
<td>BXI</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aries</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>CAPI</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intel</td>
<td>OPA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Huawei (ARM)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sunway</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Post-K (ARM)</td>
<td>Tofu</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

available | planned | OEM | unavailable

Lacks only a good GPU 😊
And what about Quantum Computing

- Potential to solve difficult problems
  - Classical bit VS Qubits
- Only a few "real" quantum computers
  - Mostly simulators
  - Ex: QLM (ATOS + partners)
- Develop new algorithms
  - The "logic" is not the same
- Designing computing architectures
  - Many challenges on memory access, interconnection
Quantum as accelerator

• Quantum gates can speed up some complex operations
• Ex: Hadamard Gate
  • Equivalent to a Discrete Fourier Transform
  • Can replace part of complex operations and outperform classical algorithms

quantum.gov/pmc/articles/PMC2596249/

quantum.gov/pubs.acs.org/doi/10.1021/acscentsci.8b00788

Schrodinger

• SAll, requires dozens of qubits

CHAPTER 4. QUANTUM COMPUTING

Hadamard Gate

The Hadamard gate is the first authentic quantum gate because it has the capacity to generate superposition states. It maps the basis states as follow:

<table>
<thead>
<tr>
<th>0_i</th>
<th>0_i + 1_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1_i</td>
<td>1_i - 1_i</td>
</tr>
</tbody>
</table>

This implies that a measurement will have equal probabilities to become 1 or 0. It represents a rotation of \( \pi \) about the axis \( \hat{x} + \hat{z} \) / \( \sqrt{2} \). This is the combination of two rotations: \( \pi \) about the Z-axis followed by \( \pi/2 \) about the Y-axis. Its matrix representation is:

\[
H = \begin{pmatrix}
1 & 1 \\
1 & -1
\end{pmatrix}
\]

The Hadamard gate is the one-qubit version of the quantum fourier transform [42]. This is extremely useful for performing the first computation in any quantum program because it transforms initialized qubits back into their natural fluid state in order to leverage their full quantum powers [32].

Square Root NOT Gate

This gate maps the basis states as follow:

<table>
<thead>
<tr>
<th>0_i</th>
<th>1_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1_i</td>
<td>0_i</td>
</tr>
</tbody>
</table>

This is extremely useful for performing the first computation in any quantum program because it transforms initialized qubits back into their natural fluid state in order to leverage their full quantum powers [32].

Table 4.2: Multiple Qubits Quantum Gates

| Quantum Computing |
| 5 to 10 years, minimum |

Hype Cycle for Emerging Technologies, 2018

Quantum Computing 5 to 10 years, minimum

gartner.com/SmarterWithGartner

Source: Gartner (August 2018)
© 2018 Gartner, Inc. and/or its affiliates. All rights reserved.
Some Conclusions and Forethoughts

- After a calm period, HPC is facing a new "Cambrian explosion" due to hardware heterogeneity
- HPC software is still bound to 2000's methods → not enough!!!
- GPUs have driven developers towards a risky path
  - Architecture-dependent
  - Low-level programming
  - This has a price
- AI is not the "holy grail"
  - Neither Quantum Computing

The next years will be agitated!
Thanks!