COMPUTING ELEMENT EVOLUTION AND ITS IMPACT ON SIMULATION CODES

OCTOBER 14TH, 2014

P(ND)^2-2

Guillaume Colin de Verdière

P(ND)^2-2
Abstract:
This talk will cover the evolution of processing elements we will have in the forthcoming supercomputers. From this technological survey we will explain the impacts that technology will force on the simulation codes and why we won’t avoid them.
Computer architectures are getting more and more complex

- Node architecture is more and more complex
  - Reduce the number of MPI tasks
    - Increasing number of nodes (>= 5000)
- Hybrid Architecture
  - Vectorization
  - Multithreading
  - High core count
  - Multiples NUMA effects
  - Little memory per core
  - Collectives operations are expensive
    - More asynchronous ops
    - Increasing bandwidth
    - But same latency
- Compute node
  - User Code
    - process
      - Local memory
    - process
      - Local memory
    - process
      - Local memory
  --network
    - CPE
      - Local memory
    - CPE
      - Local memory

P(ND)^2

CEA, DAM, DIF, F-91297 Arpajon, France | October 14th, 2014 | PAGE 3
Constraint: dissipated power: $P = P_d + P_s$

- $P_d = C_e \times F \times V^2$ dynamic
- $P_s = V \times I_f$ static

- **To limit $P$**
  - Limit/Reduce voltage
    - ex: Pentium 4 = 1.7V; Nehalem = 1.247V
  - Limit/reduce frequencies
    - Nehalem = 2.8GHz, GPU = 1.1GHz

- **To increase compute power at constant thermal power**
  - Increase compute core count

- **Consequence**
  - Your program must be parallel!
More performance = more cores

• Options are coming from the market constraints and from the history of the providers

Hybrid computers are on the rise
Top500: 8 hybrids in the top 20
  • #1 et #2 are hybrid machines

• An early try: Road Runner
  Use the processor of a game console

• The « GPU » way NVIDIA, AMD
  Comes from a specific world: game and graphics
  • An opportunity to broaden its market shares
  A big number of simple cores (one core = 1 pixel)

• The « Manycore » way INTEL
  Comes from a general purpose world
  • Capitalize on the existing software ecosystem
  A “managed” amount of cores of mid size power
• #1 of June 2008 TOP 500
  - 1,026 petaflop/s
  - Should have been #36 top500 June 2014!

• 6562 AMD dual-core Opteron
• 12240 Cell Broadband Engine
• First Hybrid Architecture

• Large programming effort to reach performance
The « GPU » track

- Track of NVIDIA & AMD
- Working on pixels pushes massive parallelism
  - Relies on programs using a very large number of threads
  - The hardware introduces constraints on the programming style
    - E.g.: branches, synchronizations…
  - Can be seen as a big vector machine (data parallel)

- Potentially large performance
  - Yet requires a special programming of suitable algorithms
  - Still need a CPU and memory (usually twice the size of the GPU one) and an external link (PCIExpress, NVlink)
    - Incurs additional costs (big ones)
    - Emphasizes the importance of data locality
    - Increases the cost of data movements
- Some success stories at CEA
The « Manycore » track

- **Track followed by Intel**
  - “Keeps” the compatibility with the leading X86 architecture
  - Claim a simplicity of programming
    - Intel follows OpenMP standard only
  - Considered by Intel as a serious milestone on the road towards the Exascale
    - Evolution of the entire software ecosystem to follow

Tianhe-2, NUDT, China
- #1 Top500 06/14
- 33.86 petaflops
- 32000 Ivy-Bridge
- 48000 Xeon Phi (KNC)
Large core counts are getting mainstream

- **MPPA from Kalray**

- **Tilera**
  - 100 cores

- **Adapteva Epiphani**
  - 64 cores
• **Innovative architecture**
  - 72 cores + 4 threads / core
  - **Unavoidable multithreading**

• **Introduction of the stacked memory [by HMC]** (B/W ↑, latency ≡)
  - Will have a big Impact on codes (control array placement in memory)

• **2 vector units for each core**
  - Will deliver most of the compute power of the KNL
  - **Vector programming is mandatory**

• **Internal NUMA effects will appear**
Fusion architecture by AMD
Tegra architecture by NVIDIA

• **Goal**: put the GPU closer to the CPUs
  - No more communications via the PCI-Ex
  - Easy resource management for the code developer

• **Constraints**
  - Required modification to the O/S
  - Need a software ecosystem tailored to these architectures
  - **Which standards?**
    - AMD favors OpenCL, OpenMP 4.0
    - NVIDIA favors CUDA, OpenACC

• **Solutions not yet applicable to HPC**
  - Primary market = laptops, tablets, smartphones
  - Fast evolution to come
Remark: for Intel, classical processors have this kind of architecture (already)

- The Integrated Graphic Processor (IGP) is very efficient for OpenCL (1.2) programs
  - Haswell desktop
- Intel favors OpenMP 4.0
  - The software ecosystem is large
Concept of an Exaflop machine (NVIDIA)

- Dense assembly of specialized components
  - LC = Latency core: classical CPU for sequential parts
  - SM = Symmetric Multicore: SIMT engine (GPU type) for massively parallel operations
  - NoC = Network on Chip: coherent data exchange between functional units

- The question of the software (standards) still remains
Conclusions
The big trends

**Hardware**
- Reduce the electrical consumption of the compute units
  - Ever more cores with a reasonable frequency
  - Systematic usage of specialized units SIMD/SIMT

- Reduce the cost of data movements
  - Unavoidable placement of CPUs and powerful SIMD units inside the same chip
  - Integration of high performance network interfaces to the chip (SoC)
  - Stacked memory to increase bandwidth
  - More threads to hide latencies

**Software**
- Necessary evolution of standards to cope with hardware evolution
  - Don’t rush on new fancy options, yet learn by experimenting on prototypes

- Upgrade of the full software ecosystem
  - Compilers, debuggers, profilers

**Challenges**
- Programming those processors at more than 10% of their peak
- Optimize existing codes to fully utilize the platforms

---

**MPI as usual**
(PGAS maybe one day?)