DE LA RECHERCHE À L'INDUSTRIE

cea

COMPUTING ELEMENT EVOLUTION AND ITS IMPACT ON SIMULATION CODES

P(ND)^2-2 2014| Guillaume Colin de Verdière

OCTOBER 14TH, 2014

P(ND)^2-2

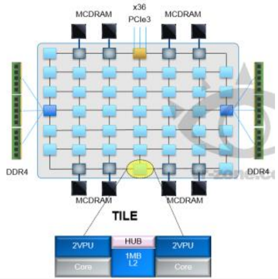
www.cea.fr

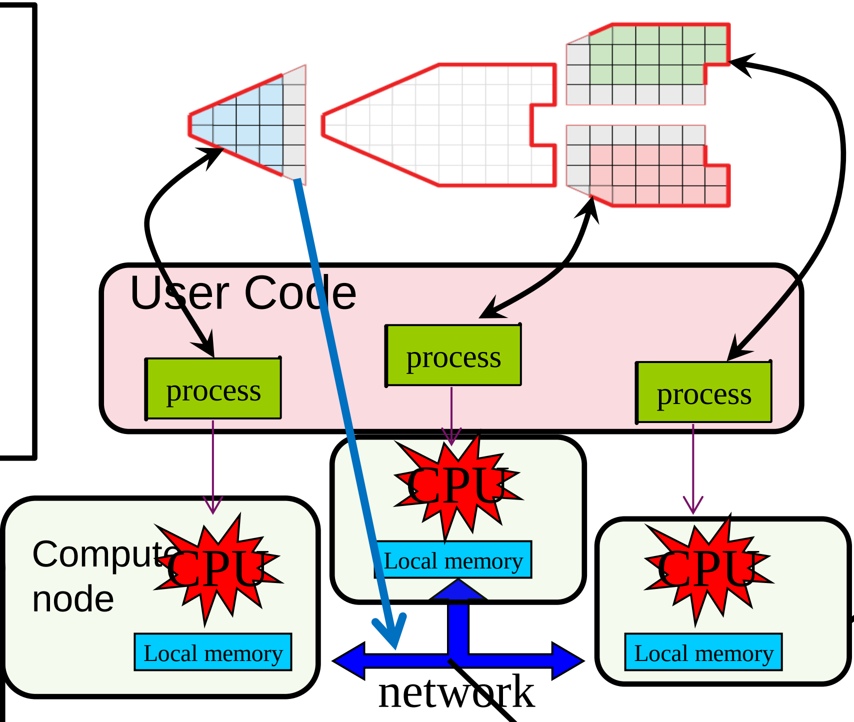
Abstract:

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

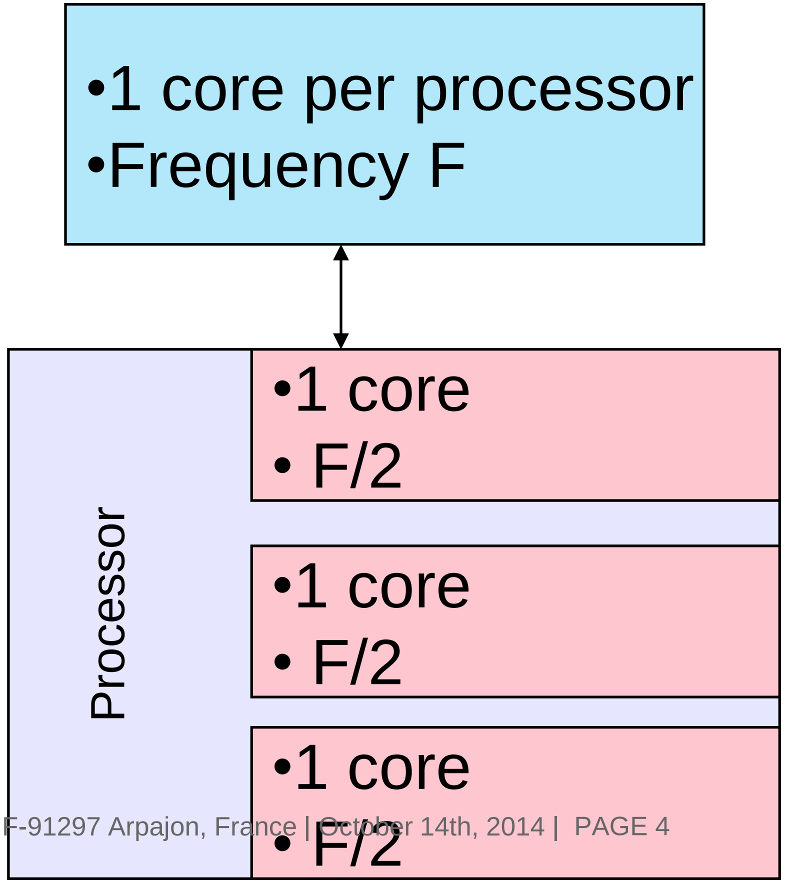
DE LA RECHERCHE À L'INDUSTRIE

Constraint : dissipated power : $P = P_d + P_s$

- I_f : leakage
- C_e : capacity of the material
- V : voltage
- F : frequency (is a function of V too)

- $P_d = C_e \times F \times V^2$ **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!**



•1 core per processor
•Frequency F

Processor

•1 core
• F/2

•1 core
• F/2

•1 core
• F/2

More performance = more cores

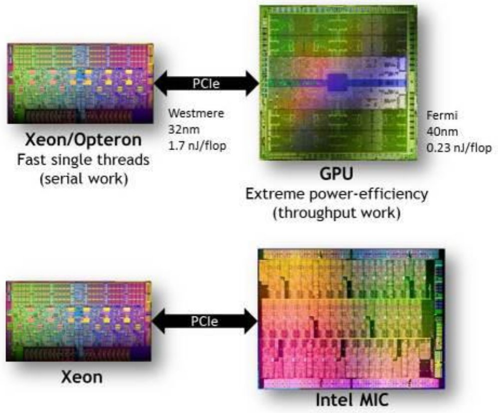
- Options are coming from the market constraints and from the history of the providers

- 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)

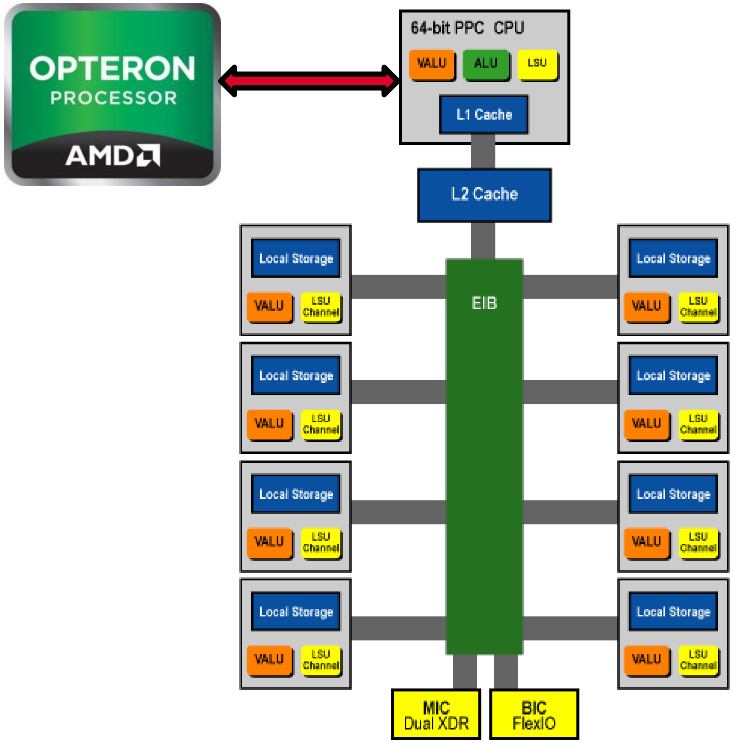
- The « Manycore » way INTEL
- Comes from a general purpose world
  - Capitalize on the existing software ecosystem
- A “managed” amount of cores of mid size power



Road Runner (IBM) @ LANL

- #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



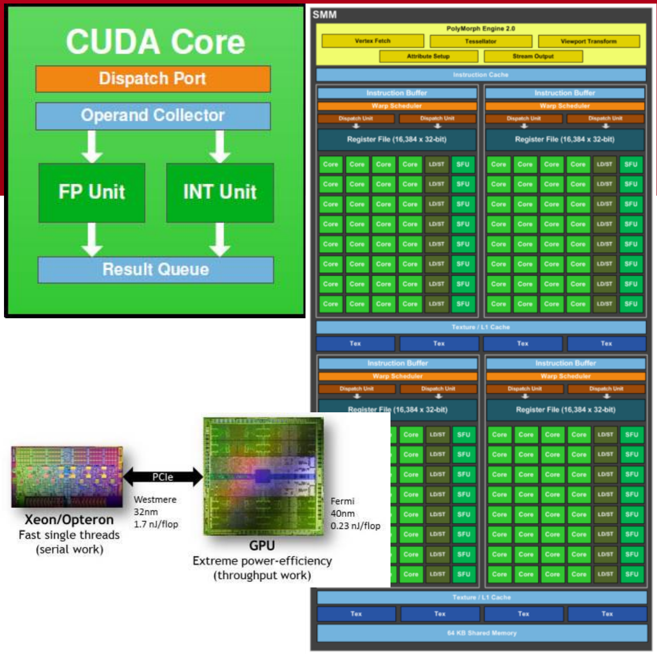
The CELL Architecture

- 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



RÉGION CHAMPAGNE ARDENNE

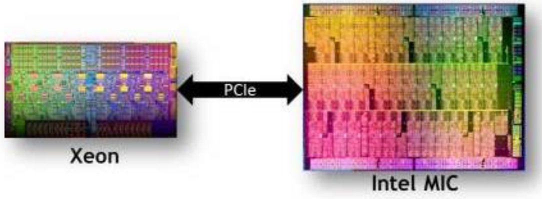
The ROMEO machine @ URCA
#184 Top500 06/14(384TFlop/s)
#6 Green500
NVIDIA K20X

The « Manycore » track

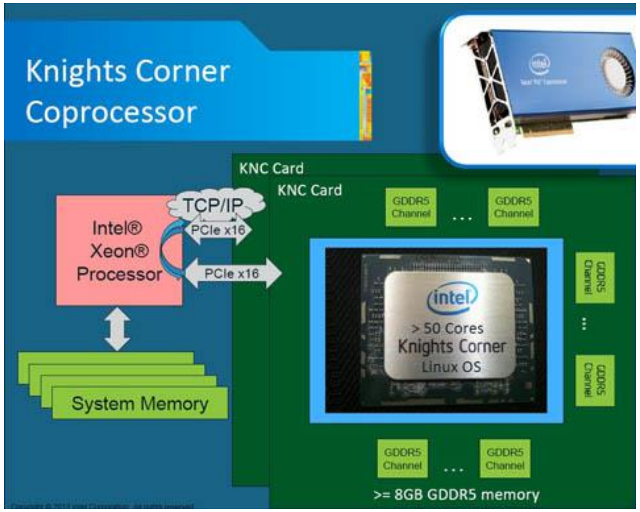
- Track followed by Intel

- “Keeps” the compatibility with the leading X86 architecture

- Claim a simplicity of programming
  - Intel follows OpenMP standard only



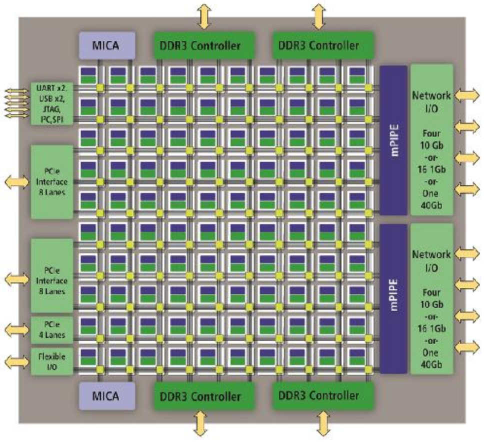
- Considered by Intel as a serious milestone on the road towards the Exascale
  - Evolution of the entire software ecosystem to follow



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

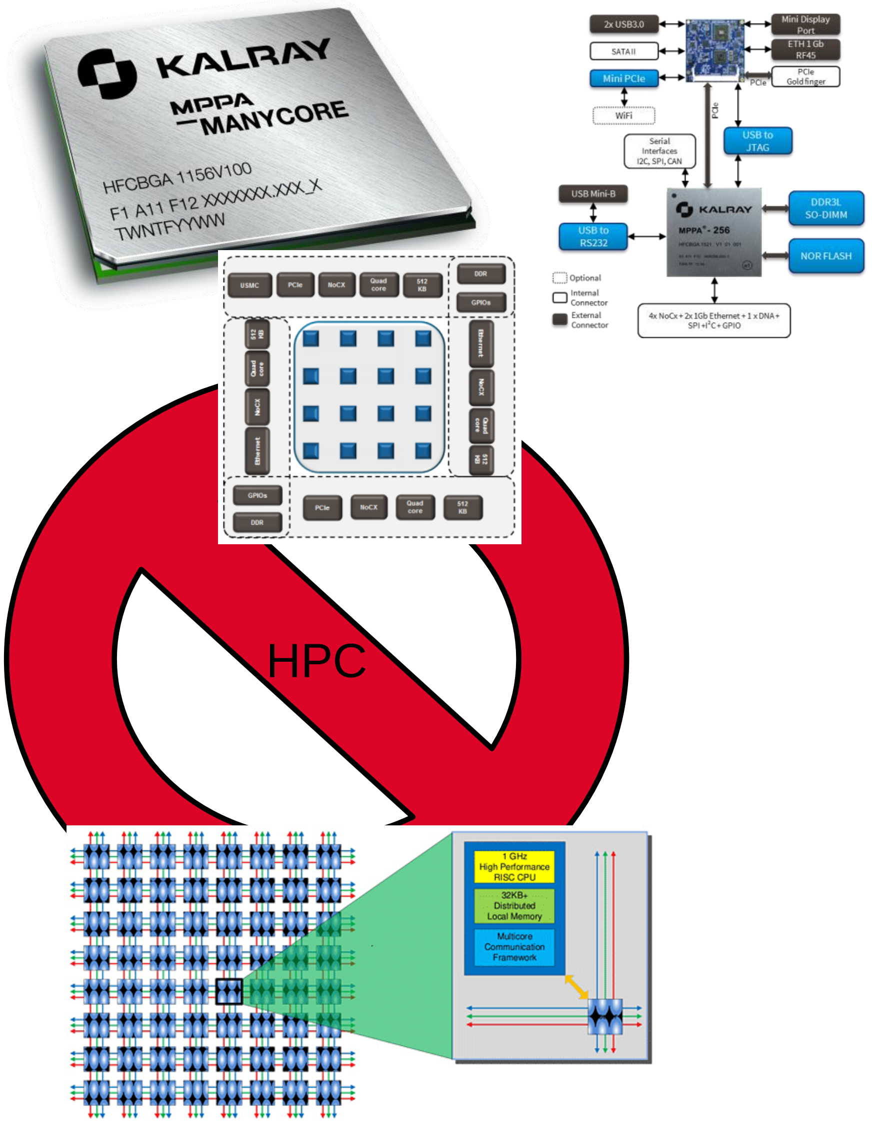


- Tilera

🟩 100 cores

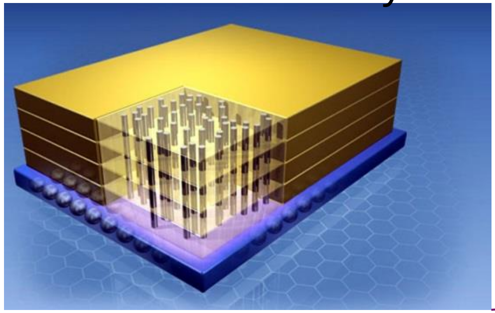
- Adapteva Epiphani

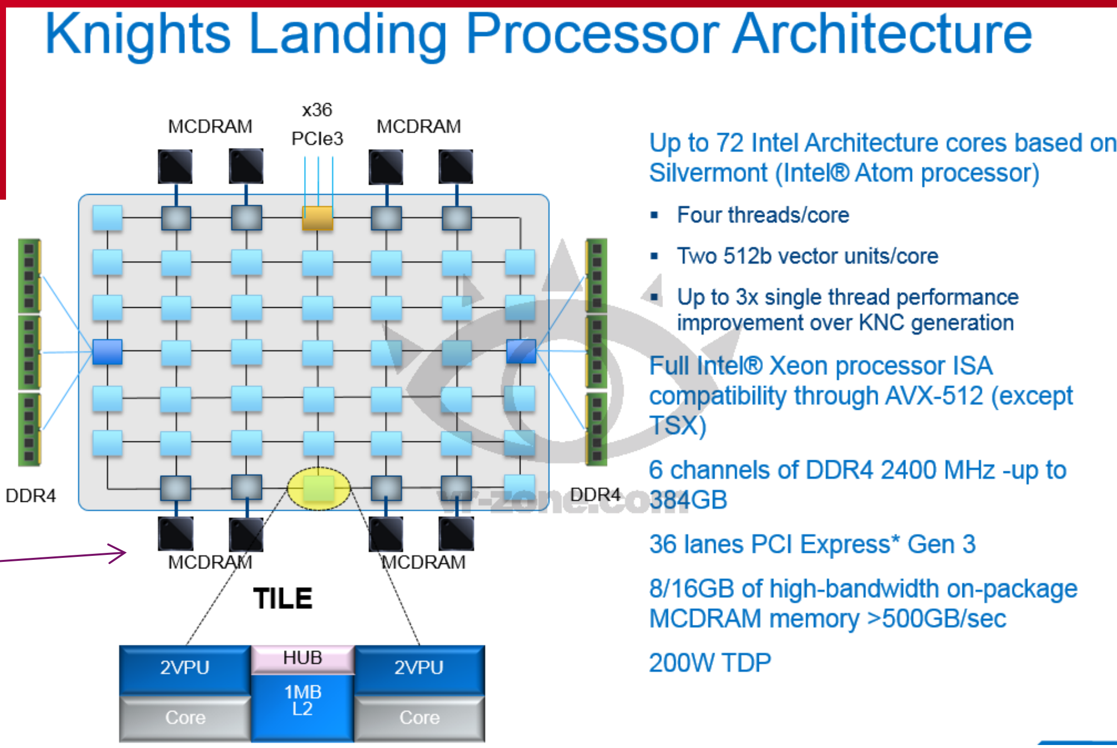
🟩 64 cores



Xeon Phi Manycore KNL (2015)

Stacked memory





Knights Landing Processor Architecture

Up to 72 Intel Architecture cores based on Silvermont (Intel® Atom processor)

- Four threads/core
- Two 512b vector units/core
- Up to 3x single thread performance improvement over KNC generation

Full Intel® Xeon processor ISA compatibility through AVX-512 (except TSX)

6 channels of DDR4 2400 MHz -up to 384GB

36 lanes PCI Express* Gen 3

8/16GB of high-bandwidth on-package MCDRAM memory >500GB/sec

200W TDP

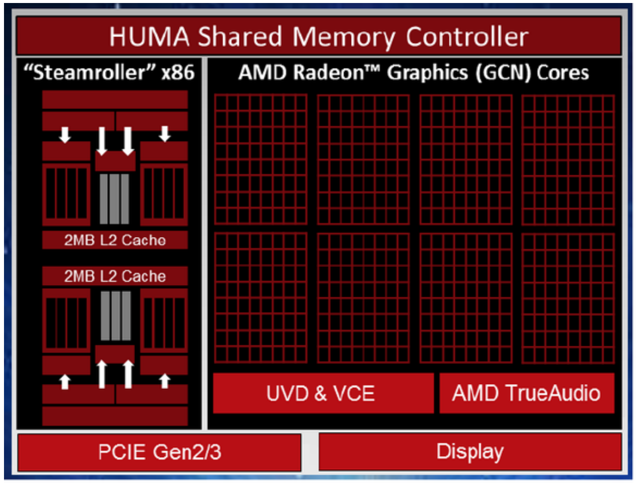
- Innovative architecture
  - 72 cores + 4 threads / core
  - **Unavoidable multithreading**
- Introduction of the stacked memory [by HMC] (B/W ↑, latency ≡)
  - Will have a big Impact on codes (control array placement in memory)
- 2 vector units for each core
  - Will deliver most of the compute power 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 developper

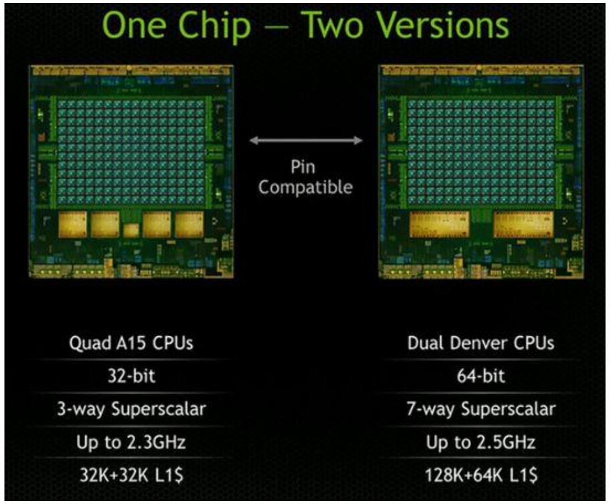
- Constraints
  - Required modification to the O/S
  - Need a software ecosystem tailored to these architectures
    - Which standards ?
      - AMD favors OpenCL, OpenMP 4.0
      - NVIDIA favors CUDA, OpenACC

- Solutions not yet applicable to HPC
  - Primary market = laptops, tablets, smartphones
  - Fast evolution to come



AMD Kaveri

NVIDIA Tegra k1

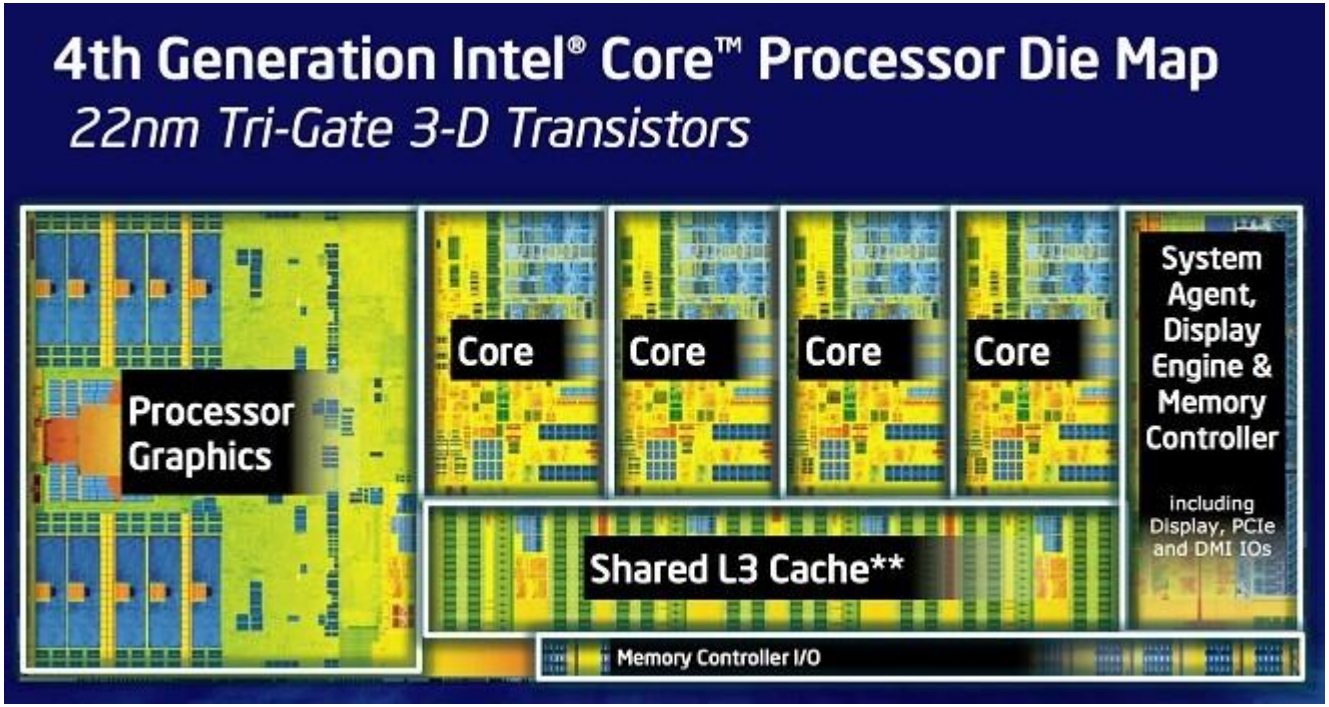


Quad A15 CPUs	Dual Denver CPUs
32-bit	64-bit
3-way Superscalar	7-way Superscalar
Up to 2.3GHz	Up to 2.5GHz
32K+32K L1$	128K+64K L1$

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

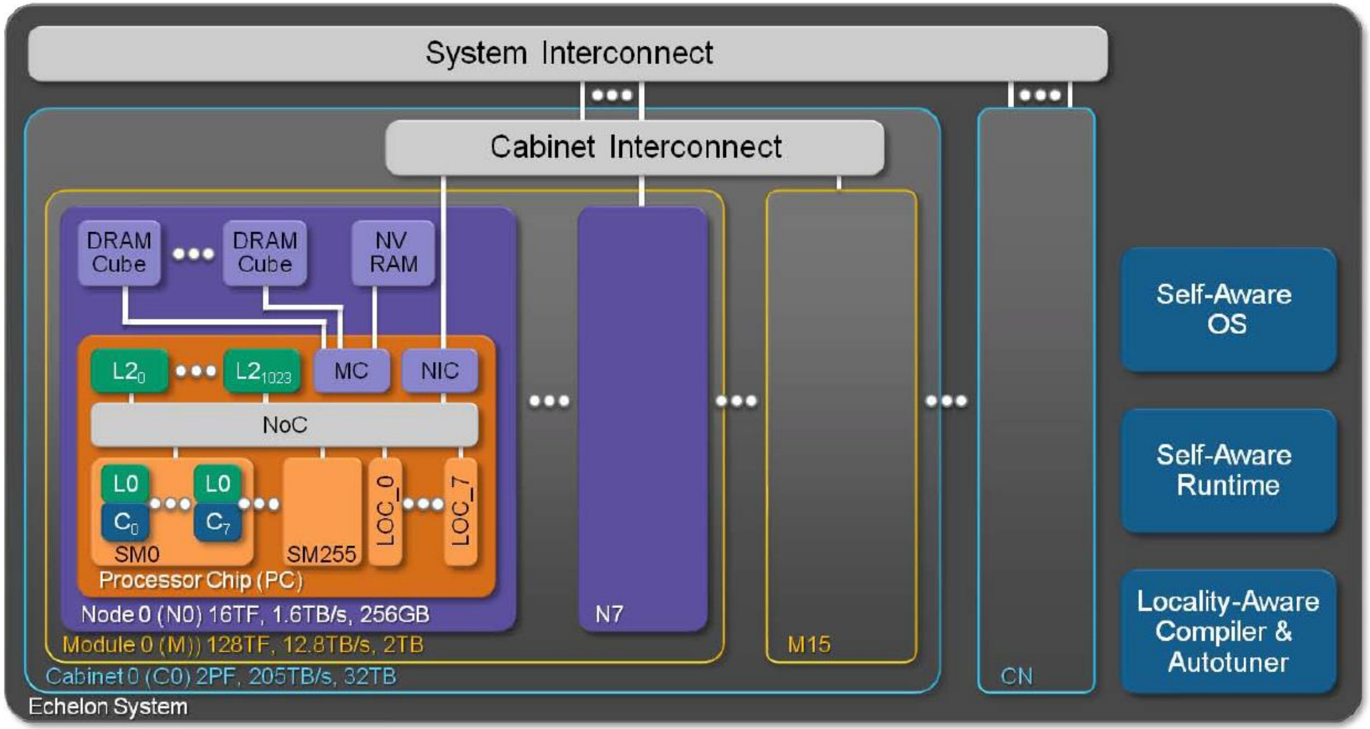

4th Generation Intel® Core™ Processor Die Map
22nm Tri-Gate 3-D Transistors
Processor Graphics
Core
Core
Core
Core
System Agent, Display Engine & Memory Controller
including Display, PCIe and DMI IOs
Shared L3 Cache**
Memory Controller I/O

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

Multithread your code

Vectorize your algorithms

Use all units

  - 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

Work on data location(s)

Multithread your code

- 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

MPI as usual (PGAS maybe one day?)

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