exdci

European
Extreme Data
& Computing
Initiative

Architecture evolutions and its impact on legacy codes

Guillaume Colin de Verdière
CEA

EXDCI-2 General Presentation

The EXDCI-2 project has received funding from the European Unions Horizon 2020 research and innovation programmed under grant agreement no. 800957.

Architecture evolutions

Evolution driver: technological constraints

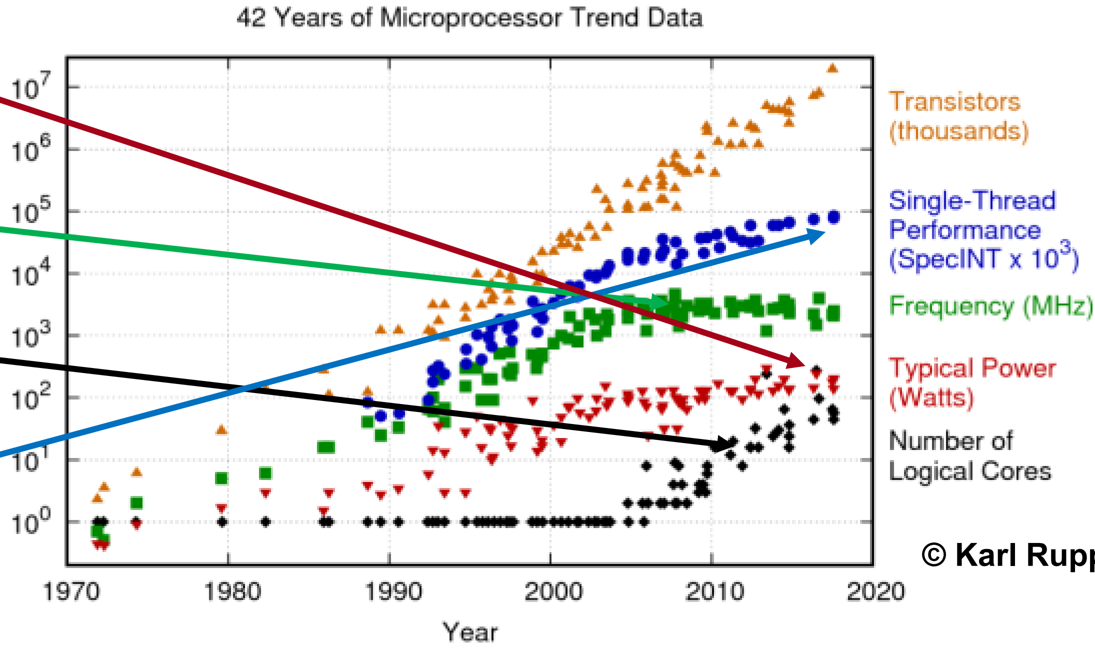
- Power wall
- Scaling wall
- Memory wall
- Towards accelerated architectures

Power wall

$$P = cV^2F$$

P: power
V: voltage
F: frequency

- Reduce V
  - Reuse of embedded technologies
- Limit frequencies
- More cores
- More compute, less logic
  - SIMD larger
  - GPU like structure



© Karl Rupp

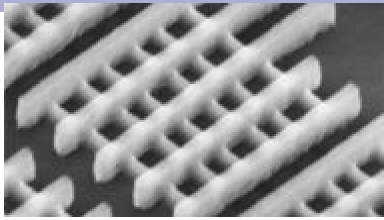
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Parallelism and vectorization are mandatory

https://software.intel.com/en-us/blogs/2009/08/25/why-p-scales-as-cv2f-is-so-obvious-pt-2-2

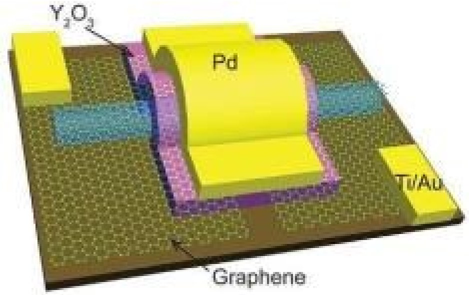
Scaling wall

FinFET



- Moore’s law comes to an end
  - Probable limit around 5 nm
  - Need to design new structure for transistors

Carbon nanotube



- Limit of circuit size
  - Yield decrease with the increase of surface
    - Chiplets will dominate
  - Data movement will be the most expensive operation
    - 1 DFMA = 20 pJ, SRAM access= 50 pJ, DRAM access= 1 nJ (source NVIDIA)

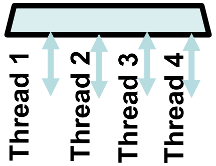
Programmers will focus on data location / movement

- **Recompute instead of storing**
- **Stay in Caches**

1 nJ = 1000 pJ, Φ Si =110 pm

Memory WALL

- Better bandwidth with HBM
  - DDR5 @ 5200 MT/s 8ch = **0.33 TB/s**
  - HBM2 @ 4 stacks = **1.64 TB/s**
- Latencies don't improve
  - More hyperthreads
- Deeper memory hierarchy
  - caches + HBM + DDR (+ NVM)
- Impact of non volatile memories?



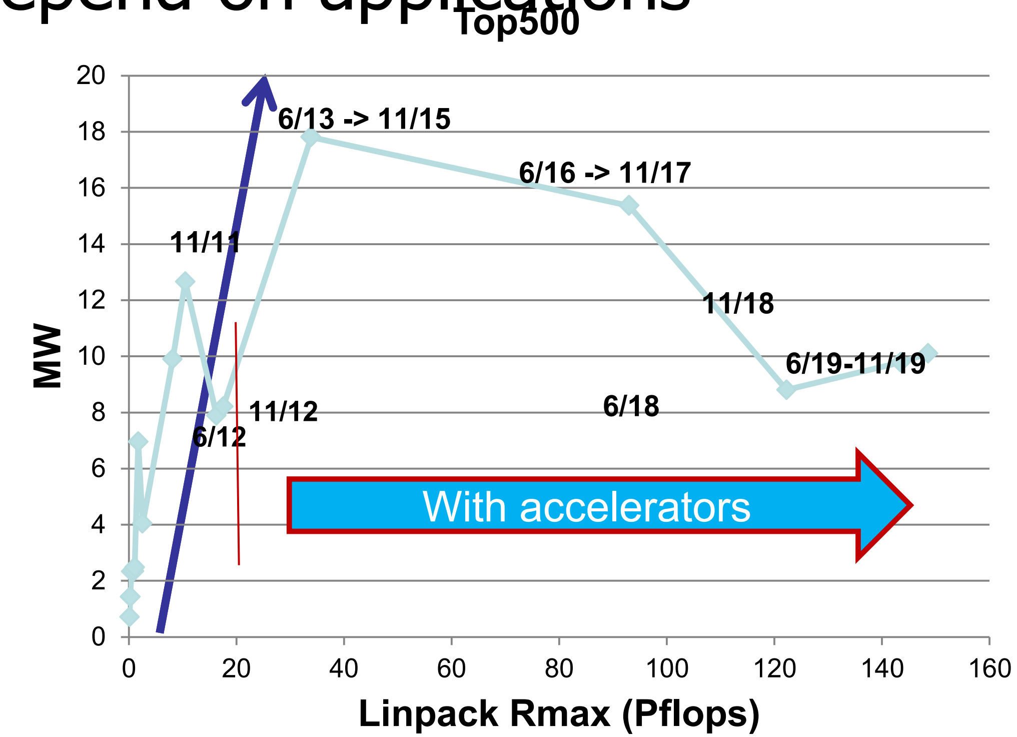
Skylake:	*SMT2*
ThunderX2:	*SMT4*
KNL:	*SMT4*
Power8:	*SMT8*

Programming will be impacted

- **Importance of spatial and temporal localities**
  - Caches will be ever more important
- **Hyperthreading is unavoidable**

Towards accelerated architectures

- Exaflop with regular CPUs will be too power hungry
- One or more accelerator per node
- Accelerator type will depend on applications
  - GPU
    - Compute and IA
  - FPGA
  - TPU
  - DSP
  - Neuromorphic
  - Quantum accelerator



Applications will have to (be) adapt(able)
Focus on data movements

Conclusion

Fundamental invariants exist

- Limit / control data movements (caches)
  - Energy & performance
- Algorithms that can vectorize (CPU and/or GPU)
  - Energy & performance
- Multi level parallelism
  - Inter node (**macro**), intra node (**meso**), vectorization (**micro**)
- *Importance of data structures and algorithms*

Impact on legacy codes

Legacy Codes

- Definition: a legacy code (LC) is a code
  - That is older than 10 years
    - Has seen at least two system generations
      - Sometimes more than 6 !
  - That is vital to the entity
  - That needs to be ported to newer systems
    - With a “reasonable” performance

Legacy code Characteristics (1/2)

- Original team (often) not available anymore
  - Need to “educate” new generations of developers
- Large source code
    - >> 1M SLOC
  - Intimate internal knowledge is potentially lost
- Needs Funding
  - to maintain the code
  - to port to the next generations of machines

Legacy code characteristics (2/2)

- Use older practices
  - Data structures ("à la CRAY")
  - Standards (F77)
- Parallelism is somewhat limited
  - MPI only, seldom GPU versions or OpenMP
- Relies on "old" third party libraries

LC: The free lunch is over

- Technology puts pressure on LC

Technology	Consequences
Energy savings	Focus on data movements
More cores & hyperthreading	More in node parallelism (threads, tasks)
Wider SIMD	Vectorize algorithms
Deeper memory hierarchy	Focus on data movements
Accelerators	Focus on data movements Vectorize algorithms

- Fast languages evolutions
  - C++/14 – 17 - …
  - Decrease of FORTRAN knowledge, increase of Python usage, …

Sutter, H. 2005. "The free lunch is over: A fundamental turn toward concurrency in software," Dr. Dobb's Journal, 30(3), http://www.gotw.ca/publications/concurrency-ddj.htm

LC: options for the future

- (not an option) stay put
- Start **rewriting** to separate « **what** » from « **how** »
  - Bottom-up approach: Frameworks (e.g. Arcane, kokkos, …)
  - Top-down approach: DSLs or task-based models (e.g. Charm++)
- Focus on performance portability for the long run
- Evaluate long term support options
  - Tools, libraries
  - Community presence

B. Hendrickson, The Day After Tomorrow: The Looming Post-Exascale Crisis, IPDPS, 2018

Legacy codes: An EXDCI-2 Vision

- Establish a quantitative measure of LC potential of evolution
  - Assumption: Exascale+ systems will be accelerated
- Code Viscosity

$$V = \frac{A * L}{TC * P}$$

  - A = age of code
  - L = SLOC in million
  - TC = Team confidence to develop accelerated codes [0.001, 1]
  - P = % of acceleration [0.001, 1]

- More to come in 2020

The EXDCI-2 project receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 800957.

Conclusion

- LC are here to stay yet they must evolve
  - A costly process – viscosity evaluation under way
    - 300 ESLOC/month = 10€/ESLOC (*)
      - ESLOC = effective source line of code, fully tested, validated, commented, documented
- Get prepared for a cultural shift
  - Team composition
  - Development methods
- Fortunately we are facing interesting times
  - Computer architectures
  - Code development
  - Algorithms and numerical methods

(* B Clark, R. Madachy, Software Cost Estimation Metrics Manual for Defense Systems, 29th International Forum on COCOMO and Systems/Software Cost Modeling) exdci

European
Extreme Data
& Computing
Initiative

EXDCI-2 General Presentation

The EXDCI-2 project has received funding from the European Unions Horizon 2020 research and innovation programmed under grant agreement no. 800957.