Argonne NATIONAL LABORATORY

Supercomputing: The Coming Decade

Marc Snir

Argonne National Laboratory &

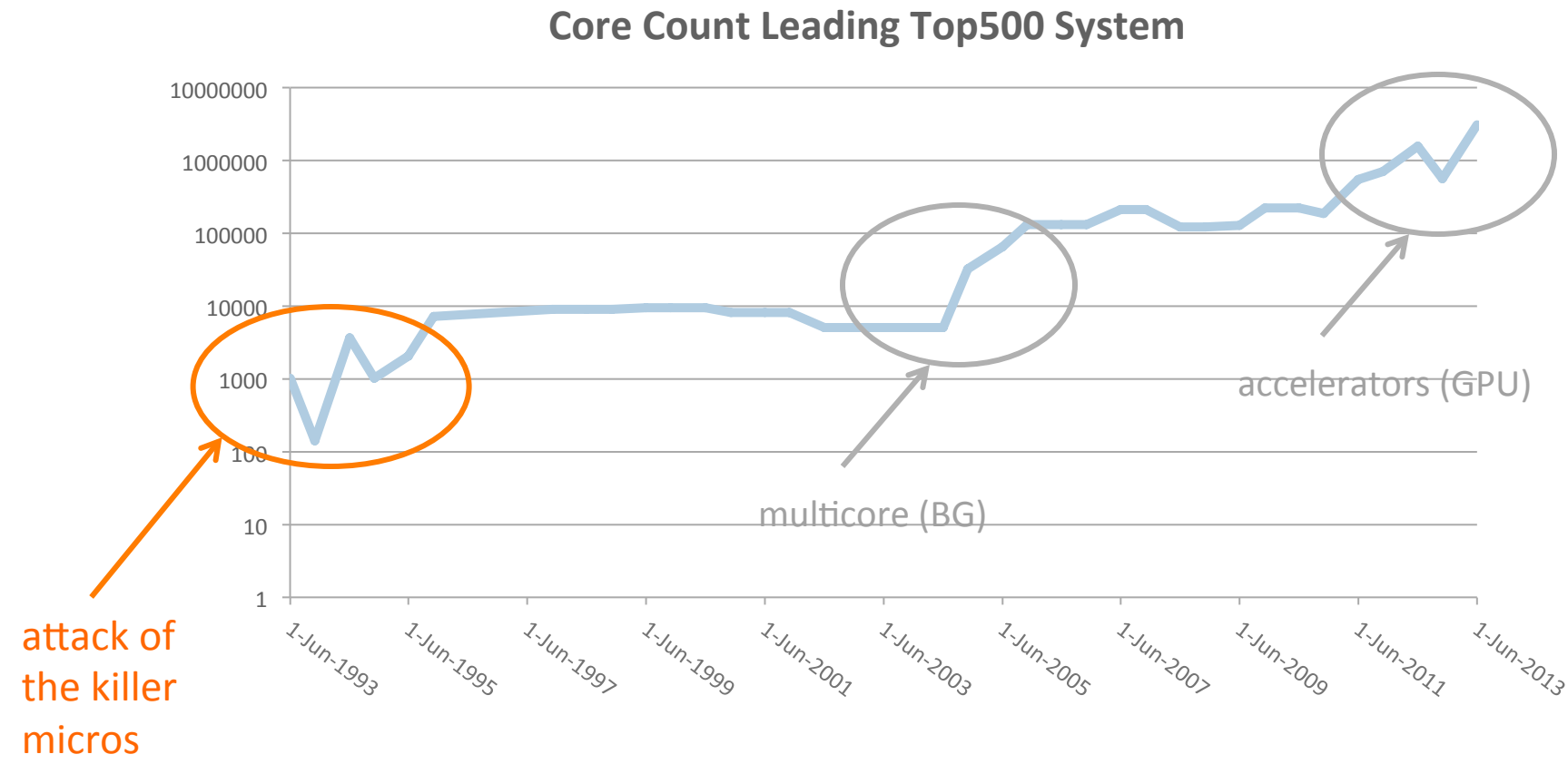
University of Illinois at Urbana-Champaign

U.S. DEPARTMENT OF ENERGY

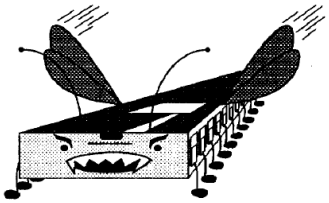
Past

Those who cannot remember the past are condemned to repeat it (Santayana)

Punctuated Equilibrium and Extinctions in HPC



The 1990 Big Extinction: The Attack of the Killer Micros (Eugene Brooks, 1990)

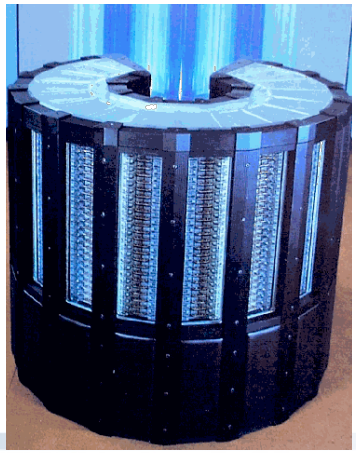


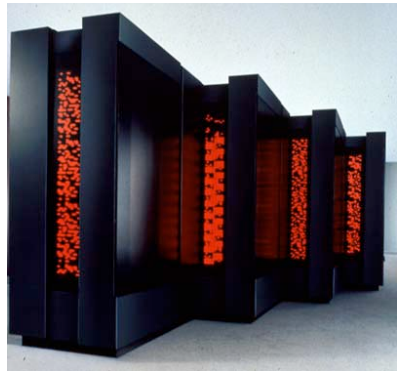
Shift from bipolar vector machines & to clusters of MOS micros

- *Roadblock*: bipolar circuits leaked too much current – it became too hard to cool them (even with liquid nitrogen)
- MOS was leaking very little – did not require aggressive cooling
- MOS was used in fast growing markets: controllers, workstations, PCs
- MOS had a 20 year history and clear evolution path ("Moore's Law")
- MOS was slower
  - Cray C90 vs. CM5 in 1991: 244 MHz vs. 32 MHz

- Perfect example of "good enough" technology

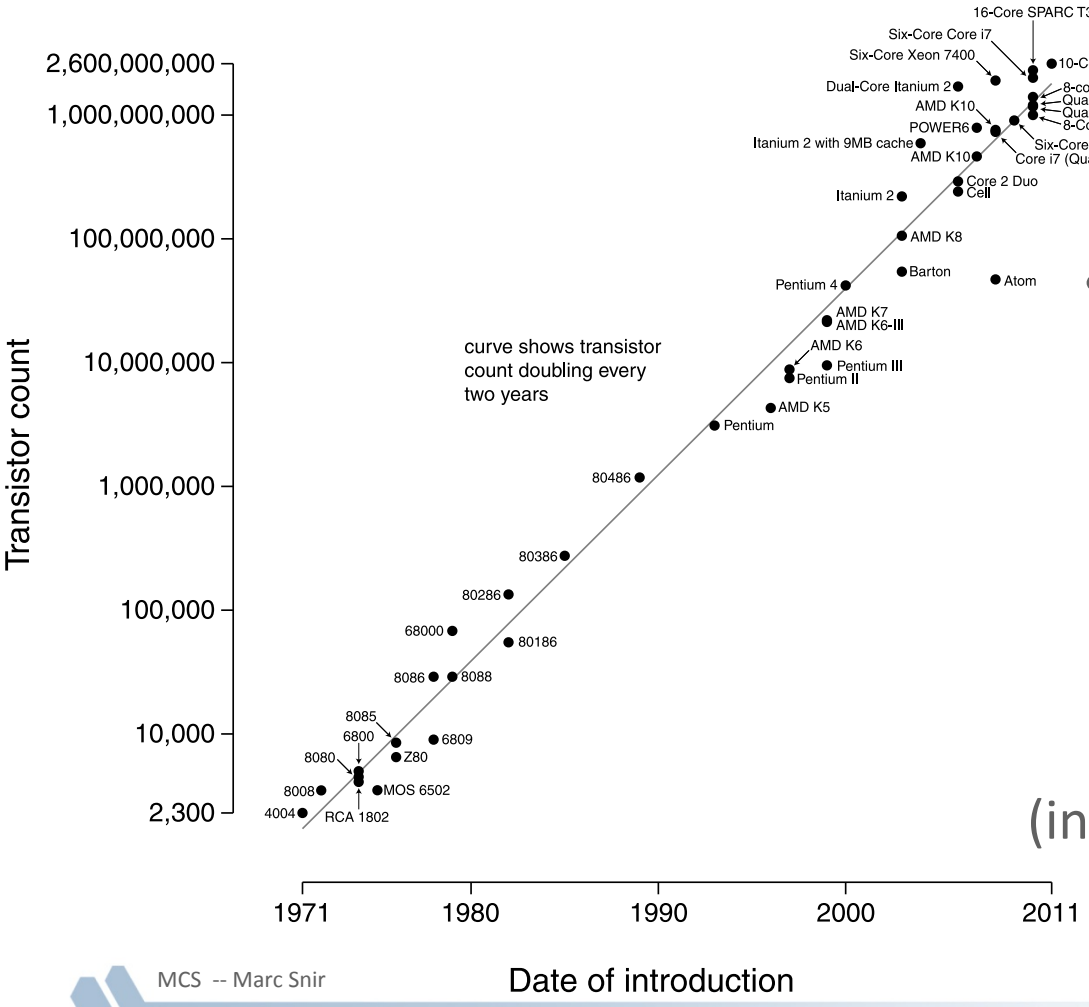
(Christensen, *The Innovator's Dilemma*)





The CMOS Age: Moore's Law & Dennard Scaling

Moore's Law: *The number of transistors per chip doubles every 2+ years*



Dennard Scaling:

- Decrease feature size by a factor of λ and decrease voltage by a factor of λ; then
  - # transistors increase by λ^2
  - Clock speed increases by λ
  - Energy consumption does not change

(in reality, voltage decrease was slower; clock speed and energy consumption increased faster)

Stasis: Moore's Law & Dennard Scaling (2)

- No change: Top number of cores on HPC system did not change for about a decade
  - Increase in performance was mostly due to increase in clock speed
- Change: Node source shifted
  - Technical workstation -> PC -> low-end server
- Change: Increased customization at the package level
  - commodity backplane -> commodity chips

Present

Stein's Law: *If something cannot go forever, it will stop*

The only question is when

Dennard Scaling Ended

- At around 130 nm in 2001-2004
- CMOS circuits leak too much current (static energy)
- Growth in density continues (multicore), but clock speed is (slowly) decreasing

- Leakage increases as feature size decreases

  While power consumption is an urgent challenge, its leakage or static component will become a major industry crisis in the long term, threatening the survival of CMOS technology itself, just as bipolar technology was threatened and eventually disposed of decades ago

International Technology Roadmap for Semiconductors (ITRS) 2011

- The ITRS “long term” is the 2017-2024 timeframe.

On Our Way to the Next Extinction?

- **History repeats itself:**
  - CMOS technology has hit a power wall
    - Clock speed is not raising
  - Alternative materials are not yet (?) ready (gallium arsenide and other III-V materials; nanowires, nanotubes)
- **History does not repeat itself:**
  - ✔ There is a much larger industrial base investing in continued improvements in current technologies
  - ✗ An alternative “good enough” technology IS NOT ready
  - ✗ There is much more code that needs to be rewritten if a new model is needed (>200MLOCs)

The Physical & Engineering Limits

- Transistor size cannot shrink forever
  - Need a few hundred atoms per gate
  - 5 nm is the limit for 2D (5 nm = 20 atoms)– might get denser with 3D
- Decreased return on feature size: Performance improvement is not proportional to size reduction
  - Additional spacing and larger safety margins needed to reduce interference, handle manufacturing variances, etc.
- Reduced leakage requires technology innovation
  - New materials (III/V, nanotubes…), 3D devices
- Need new light sources
  - Current 192 nm
- …

Technological challenges of high-performance logic scaling

Argonne

Year	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024
Gate Length	18	16.7	15.2	13.9	12.7	11.6	10.6	9.7	8.8	8.0	7.3
Equivalent Oxide Thickness	●	●	●	●	●	●	●	●	●	●	●
Source-Drain Leakage*	●	●	●	●	●	●	●	●	●	●	●
Threshold Voltage*	●	●	●	●	●	●	●	●	●	●	●
CV/I Intrinsic Delay	●	●	●	●	●	●	●	●	●	●	●
Total Gate Capacitance	●	●	●	●	●	●	●	●	●	●	●
Drive Current	●	●	●	●	●	●	●	●	●	●	●

- Time line shown for best performing multi-gate transistor technology.
- Based on ITRS reports (2011*, 2012* and 2013 eds.)

● technology available
● solutions known
● no known solutions

(courtesy Denis Mamaluy)

U.S. DEPARTMENT OF ENERGY

Notes on Scaling Challenges

- For a given EOT, MG (multigate) devices have much better gate-over-channel control than bulk/SOI devices.
- In order to compensate for the decreased gate-over-channel control for smaller gate lengths, EOT has to be reduced. Yet too small EOTs result in unacceptable **gate leakage** (tunneling) currents which also result in power loss.
- **Source-to-drain leakage current $I_{sd,leak}$** arises due to insufficient gate-over-channel control and leads to power loss and other severe adverse effects. The maximum allowed $I_{sd,leak}$ is the most fundamental technology parameter in the sense that it does not change with the gate length or other parameters and is set to 100nA/µm.
- **Supply (drain) voltage V** could also be reduced in order to decrease the power consumption and delay; however any decrease in the supply voltage would result in the corresponding decrease of the **threshold voltage V_{th}** (the gate voltage at which the device is turning on). In practice this is not permissible, since the threshold voltage is already at its limit: lowering V_{th} dramatically (exponentially) increases the source-drain leakage current.
- The main speed characteristic of high-performance logic is the **intrinsic delay τ** (the time taken by the gate to produce an output after giving the input)
- **Gate capacitance C_g** should be small in order to decrease intrinsic delay; smaller gate lengths decrease gate capacitance, yet smaller EOTs result in higher gate capacitances.
- **Drive current I_D** is thought to be the main mean to reduce intrinsic delay $\tau = CV/I$, yet its increase usually results in higher source-to-drain leakage currents.

EOT = equivalent oxide thickness
I_D = drive current
C_g = total gate capacitance
$I_{sd,leak}$ = source-drain leakage
V_{th} = threshold voltage
$\tau = CV/I$, intrinsic delay

The Economical Limits

- Cost per transistor has not decreased last year
  - Market for increased performance at increased cost is very small
- Investments for new fabs keep growing, resulting in increased consolidation
  - Some predict only two vendors will be left below 22nm
- Cost of manufacturing chips keep increasing
  - More materials, more masks, more passes
- IC market cannot grow forever faster than GDP
  - Fast growth is necessary to amortize the large investments in new fabs

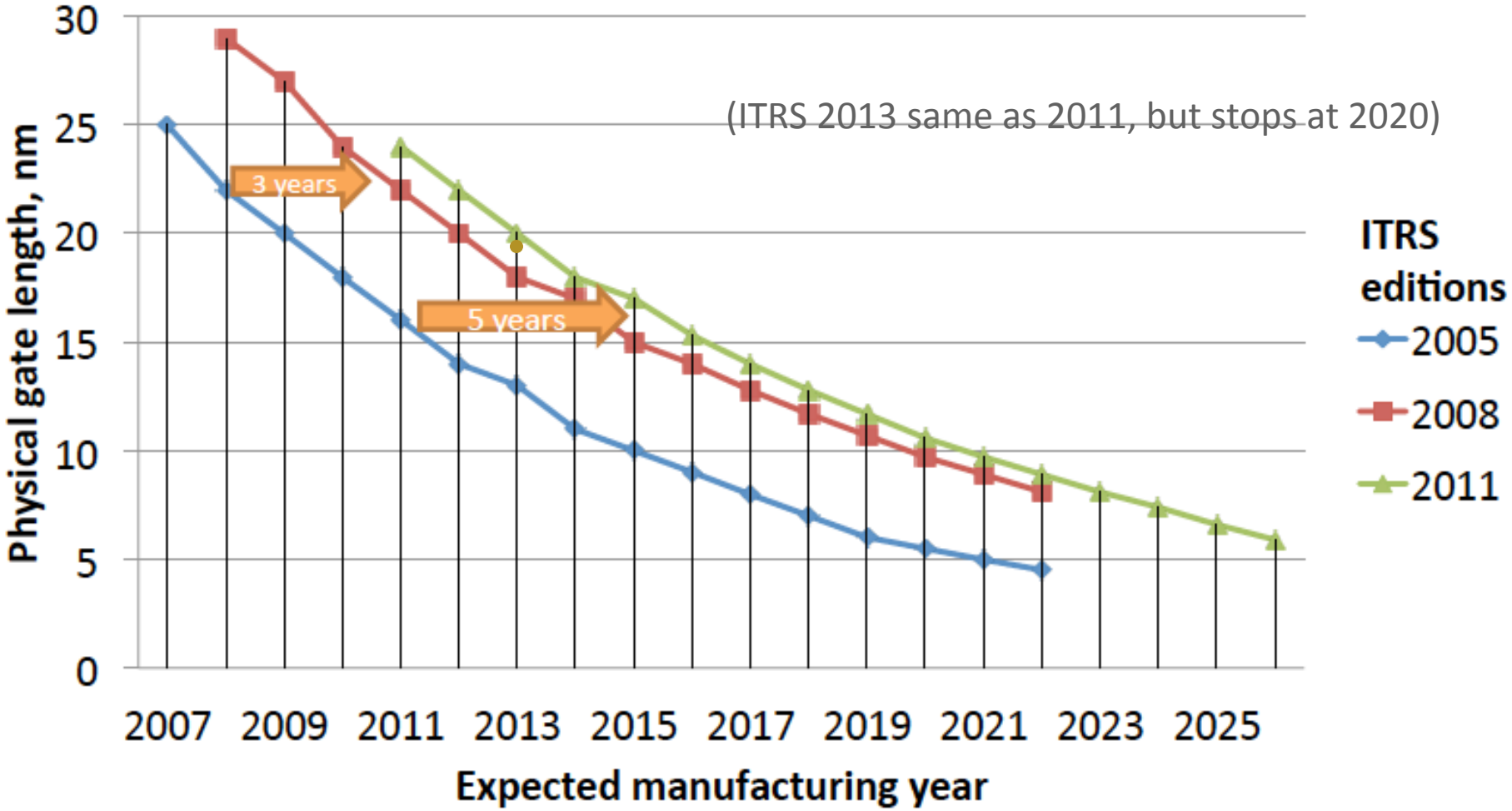
The Market Constraints

- Leading market for IC is mobile. The drivers in the market have little overlap with HPC.
  - ✔ low power
  - ✔ system on chip
  - ✗ small form factor
  - ✗ integration of analog and MEMS
  - ✗ limited interest in low error rate
  - ✗ no interest in 64 bit floating point and higher

Future

Life can only be understood backwards; but it must be lived forwards.
(Kierkegaard)

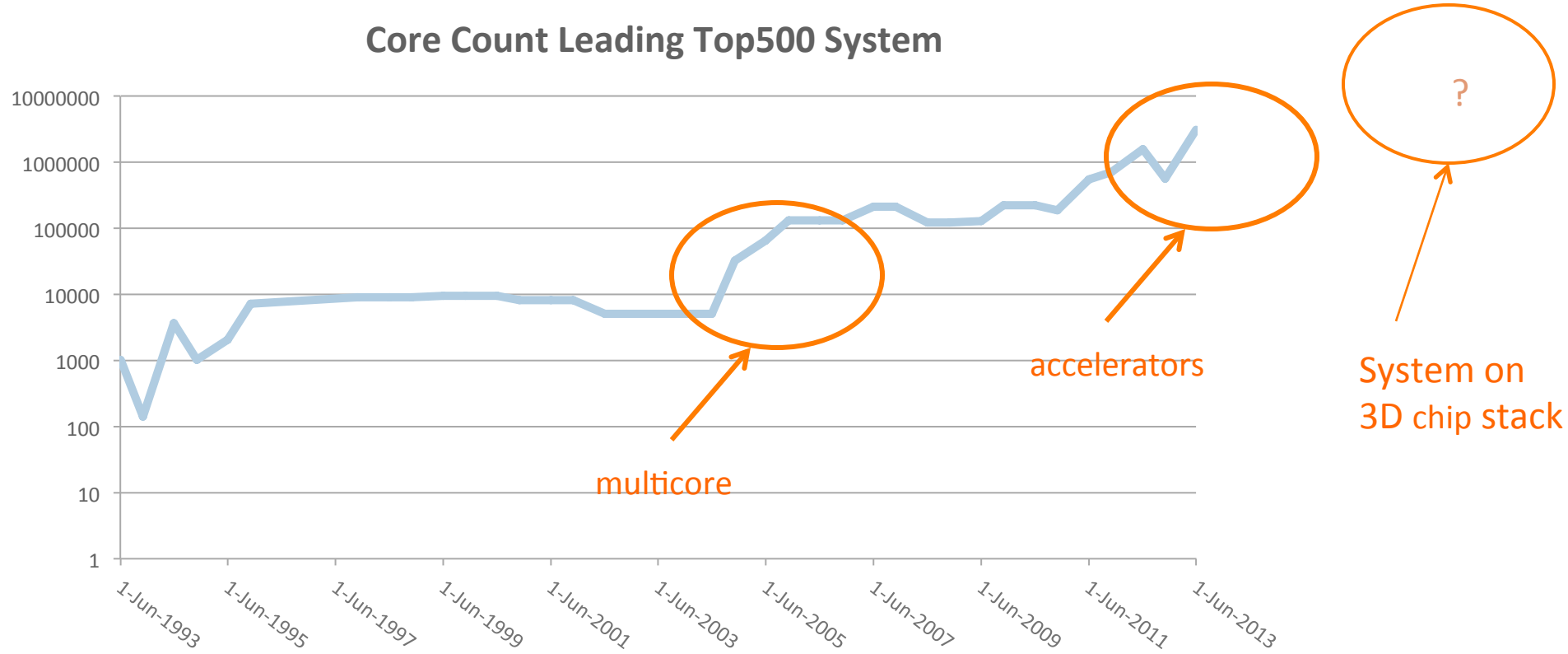
The Future Is Not What It Was



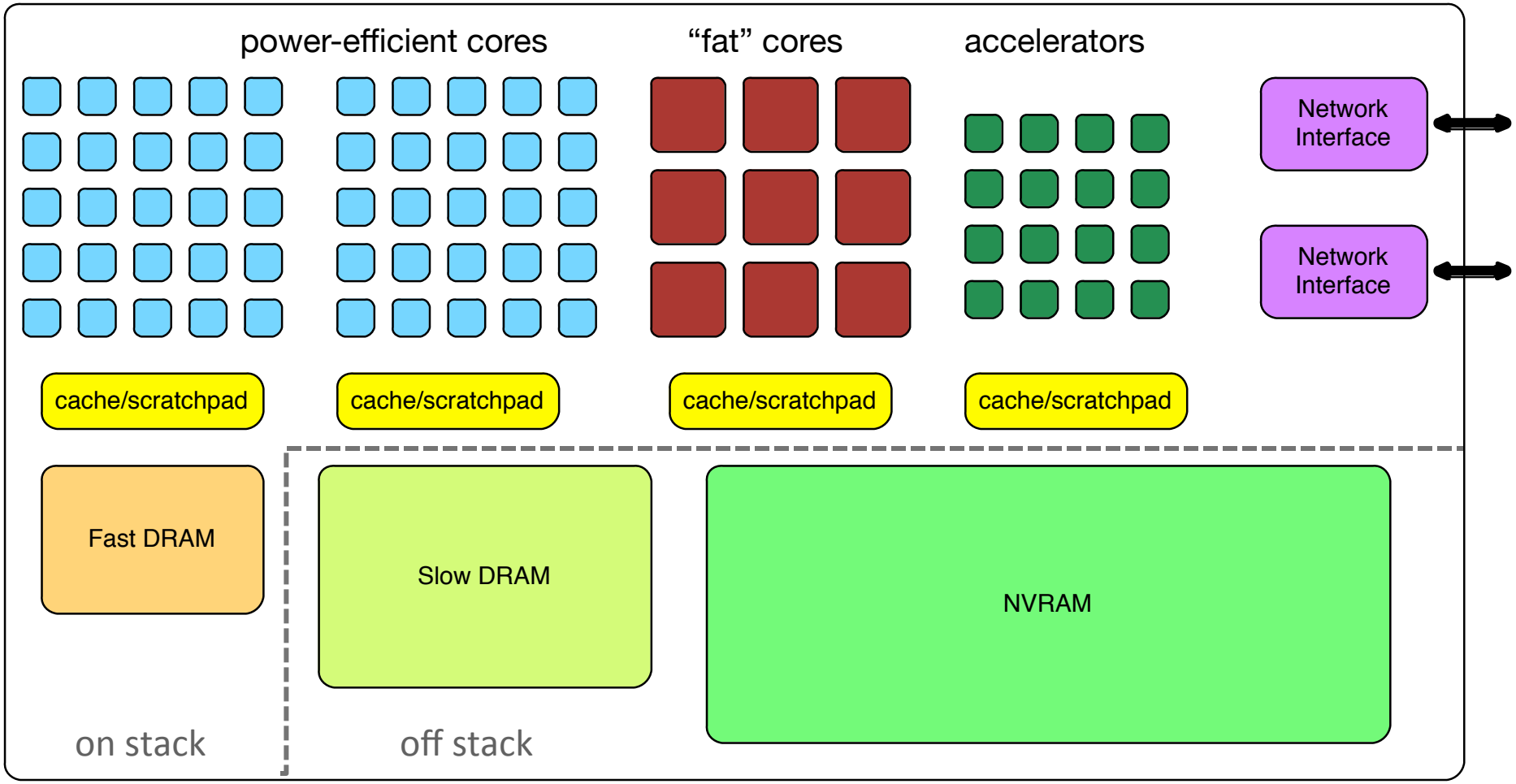
The sky is falling, but not immediately

(courtesy J. Aidun)

The Impact of the Energy Wall on HPC

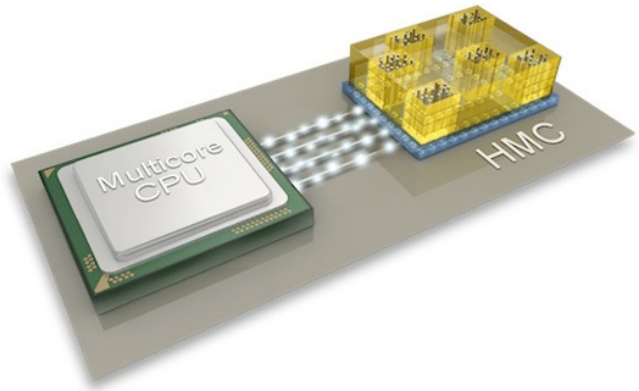


Future Node (~4-8 years)



Future Nodes (~4-8 years)

- Technology
  - 3D packages – e.g., for memory
  - NVRAM as memory replacement (rather than disk)
  - 100's of threads
  - heavily NUMA/NUCA
  - fine-grain power management
  - dark silicon
  - Silent data corruption problem?



- Economics
  - Nodes are systems on a chip – may integrate IP from multiple vendors
  - Nodes are designed for a small, specialized market (mostly HPC)
  - New vendor ecosystem?

Software Challenges

- **Scale**: billions of threads
- **Heterogeneity**:
  - Cores: throughput and latency optimized cores & accelerators
  - Memory hierarchies, including SRAM, nearby and remote DRAM, NVRAM; heavy NUMA; possibly noncoherent
- **Energy**: power as a first-class resource
- **Resilience**: frequent and possibly silent HW errors
- **Variability**: coping with continuous change and variable execution speed
- **New workloads**: Workflows, simulation+analysis, multicomponent applications
- **Complexity**!

Solutions?

- No expectation that new programming models will play a major role
  - ~200MLOCS is a huge investment
  - MPI + OpenMP, with suitable enhancements seem viable
  - No alternative programming model offers major advantages
- Enhancements:
  - MPI + 100's of threads, fault-tolerant MPI, locality-aware OpenMP
  - DSLs, frameworks and libraries
  - Use of suitable programming patterns
- *Do not seek the next programming model – design for portability*
- Algorithms will need to tolerate asynchrony
- The underlying OS and runtime services will need to change

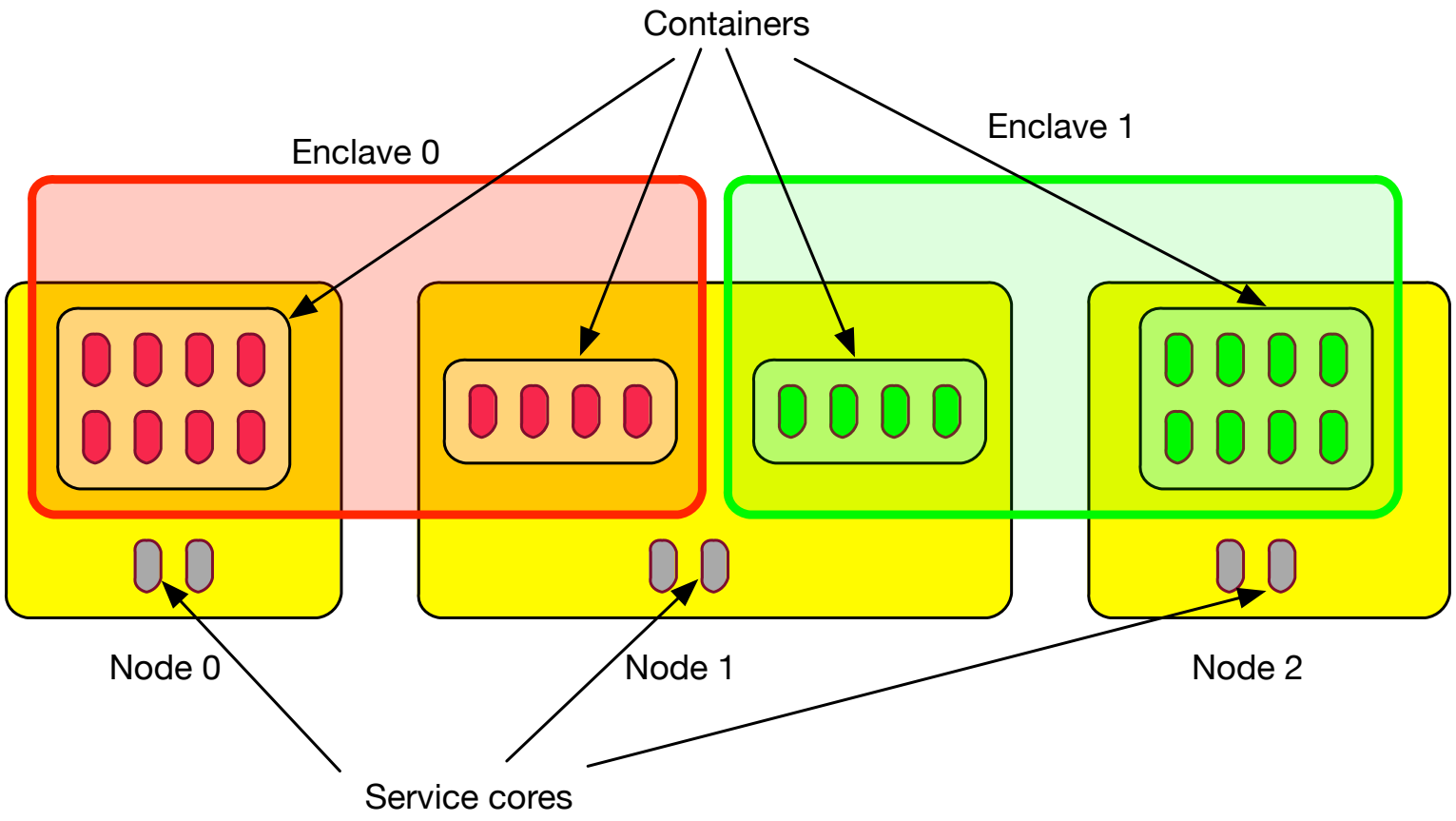
Current limitations of HPC OS and Runtime

- Resource management is machine-global and static
- No management of power or network bandwidth and only limited management of I/O resources
- No flexibility in error/fault management
- No constructs for coordinating workflows
- Unproven (at best) capabilities for managing node challenges: O(1000) threads, heterogeneity of cores, or complex memory hierarchies
- Overly simplistic definitions and mechanisms for supporting isolation

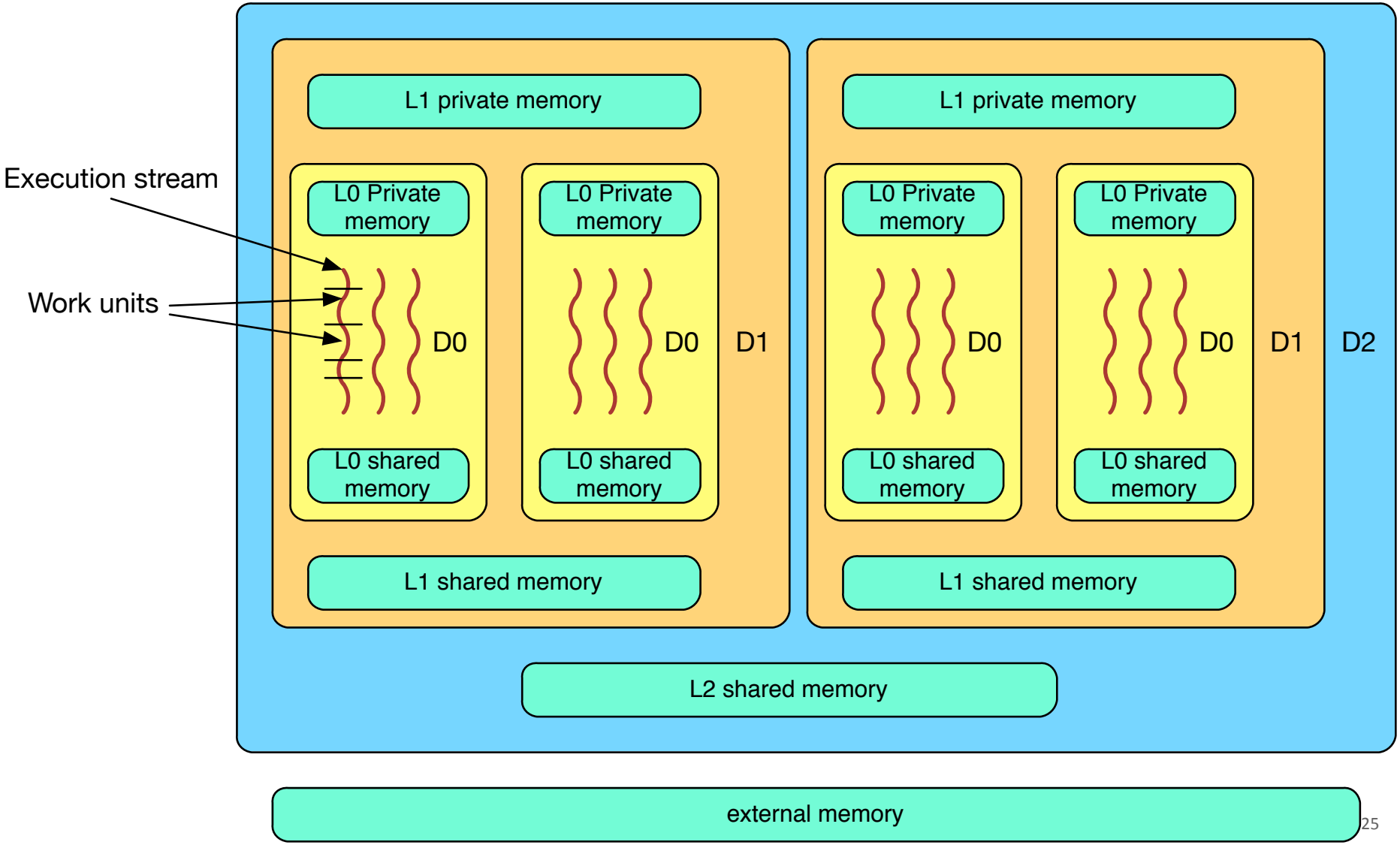
Argo, Hobbes, X-ARCC Design Principles

- Exploit **hierarchy** to enable scalability
- Manage resources in **runtime**, rather than OS
- Runtime can be **application specific**
- Support for **adaptive resource management**: hierarchical control with feedback
- **Performance isolation** (QoS) to enhance resource utilization (and avoid over provisioning)
- **Fault isolation** to support local, independent recovery
- **Customization** to support variety in software and hardware

Containers & Enclaves



Container Run-Time



Post-Moore? (10 years +)

- No “good enough” (cheap) technology is waiting in the wings
  - as distinct from the situation when bipolar reached its end
  - New materials and new structures are postponing the end of the Field Effect Transistor (FET) world, not avoiding it. Same for other proposed logic devices
- **HW and SW specialization can probably provide 1-2 orders of magnitude performance growth**
- How about something totally different?
  - Quantum computing, biologically inspired computing
  - No evidence (yet?) these models apply to scientific computing,

How to get 1-2 Additional Order of Magnitudes: Frictionless Computing

- Main constraint is energy
- Energy is spent on communication (in time and space)

2015 technology

flop = 10 pJ

(Courtesy P Kogge)

Operation	Energy per bit
Register File Access	0.16
SRAM Access	0.23
DRAM Access	1
On-chip movement	0.0187 per mm
Thru Silicon Via (TSV)	0.011
Chip-to-Board	2
Chip-to-Optical	10
Router on-chip	2

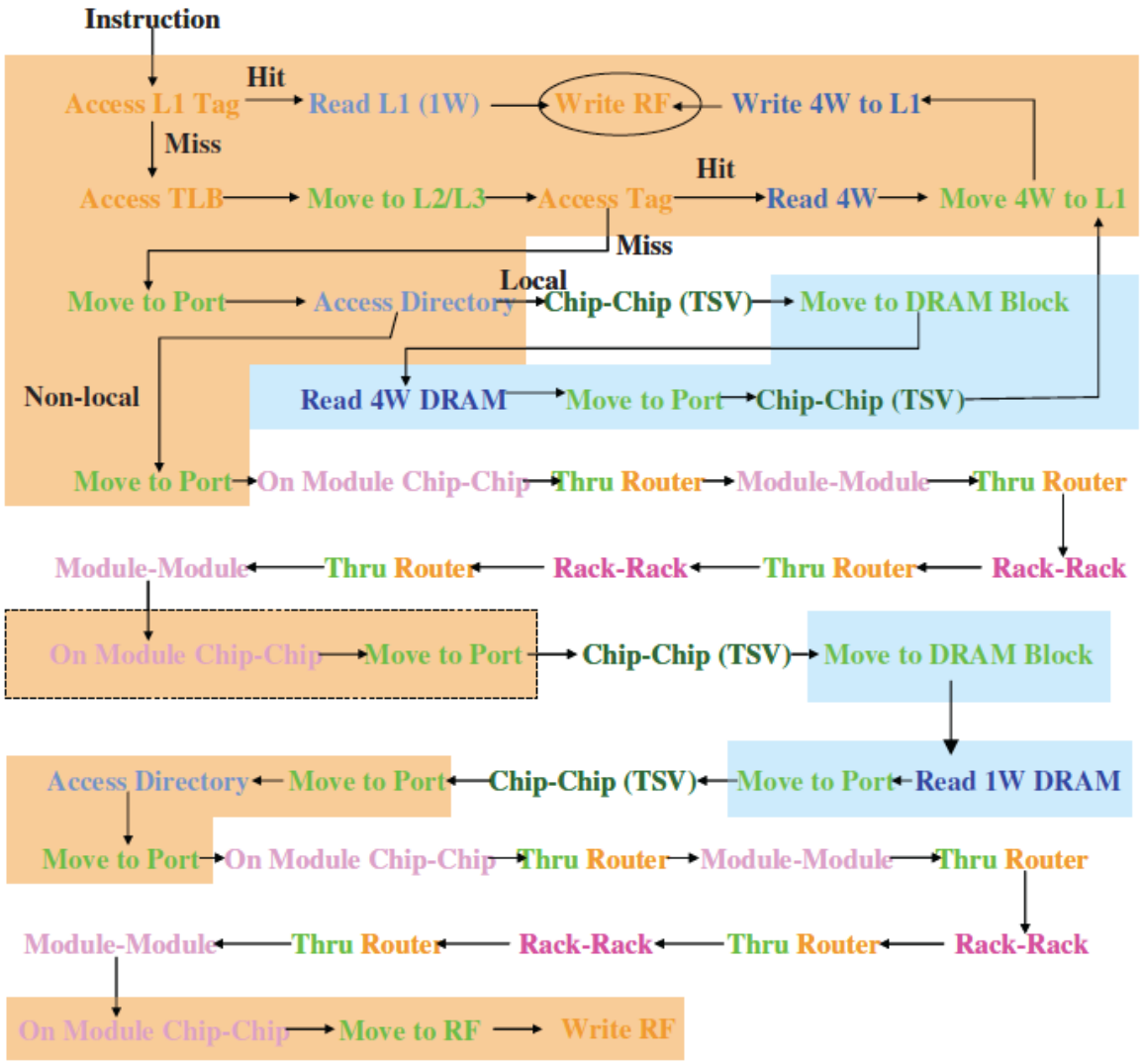
How to get 1-2 Additional Order of Magnitudes: Frictionless Computing

- pJoules spent for one flop in Linpack
  - Flop is 2% of cost!

Step	Target	pJ	#Occurrances	Total pJ	% of Total
Read Alphas	Remote	13,819	4	55,276	16.5%
Read pivot row	Remote	13,819	4	55,276	16.5%
Read 1st Y[i]	Local	1,380	88	121,400	36.3%
Read Other Y[i]s	L1	39	264	10,425	3.1%
Write Y's	L1	39	352	13,900	4.2%
Flush Y's	Local	891	88	78,380	23.4%
Total				334,656	
Ave per Flop				475	

Doing it Better: Architecture + Algorithm

- Most energy is "wasted"
  - E.g., 10's of SRAM accesses in order to bring data from memory
  - Load of one memory word is x17 more expensive than it needs be!
- "Communication-avoiding algorithm" could have many fewer memory accesses.
  - n^2 vs. n^3



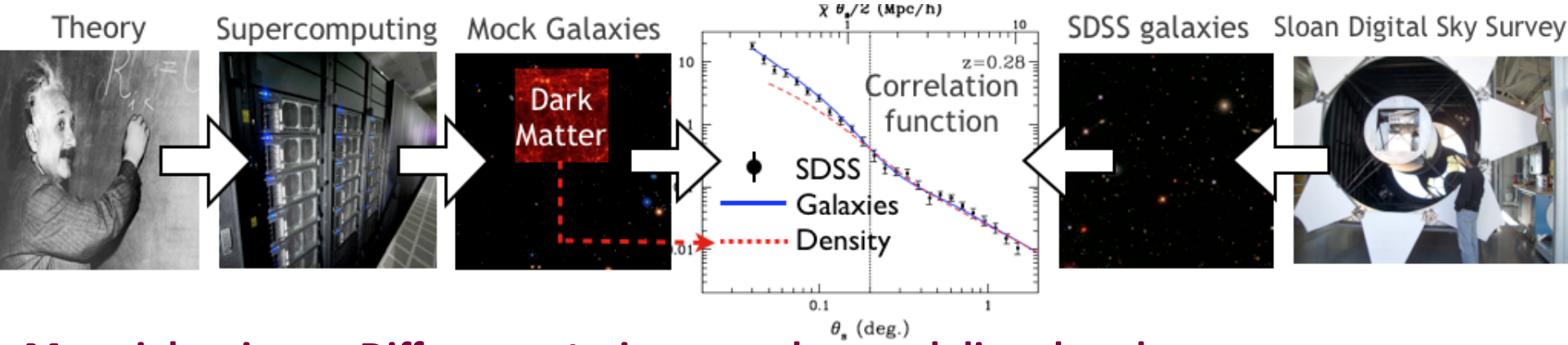
Do we Care if Supercomputing Performance Plateaus?

- It's all about Big Data Now, simulations are passé
- B***t
- *All science is either physics or stamp collecting*. (Ernest Rutherford)
  - In Physical Sciences, experiments and observations exist to validate/refute/motivate theory. "Data Mining" not driven by a scientific hypothesis is "stamp collection".
- Simulation is needed to go from a mathematical model to predictions on observations.
  - If system is complex (e.g., climate) then simulation is expensive
  - Often, models are stochastic and predictions are statistical – complicating both simulation and data analysis

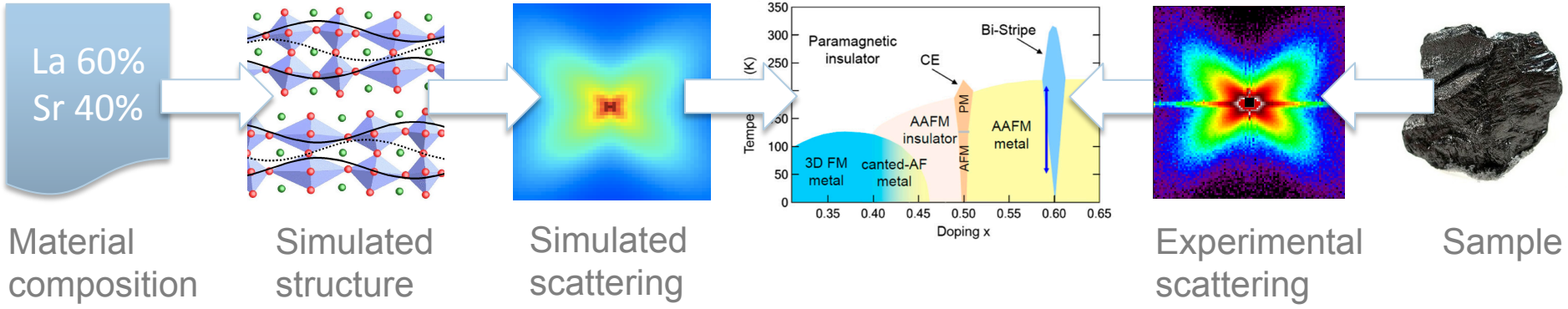
Data-Driven Science Examples

For many problems there is a deep coupling of observation (measurement) and computation (simulation)

Cosmology: The study of the universe as a dynamical system



Materials science: Diffuse scattering to understand disordered structures



Images from Salman Habib et al. (HEP, MCS, etc.) and Ray Osborne et al. (MSD, APS, etc.)

Summary

- Exascale will be there by 2022 or so
- No revolution in sight for exascale or beyond– it just becomes progressively harder to improve performance.
  - Circuit technology contributes less and less
  - Architectures and algorithms contribute more and more
- If we really want it, zetaflop/s performance will be reached in 203x.
- *It will be an international effort*
- We should really want it – supercomputers are not prestige toys
- If we go to there, the next decade will be much more interesting than the previous decades of supercomputing
- If not, life will be boring and I shall go fishing


Questions?

Performance Portability

- Scientific codes live much longer than computer systems (two decades or more); they need to be ported across successive HW generations
- Amount of code to be ported continuously increases (major scientific codes each have > 1MLOCs; total estimated at > 100 MLOCs)
- As Moore's law slows down, architecture will evolve more rapidly
  - May not converge to a new, stable paradigm
- Need very efficient, well tuned codes (HPC platforms are expensive)
- Need portability across platforms (HPC programmers are expensive)
- Need software technologies to bridge these conflicting forces

Resilient Massive Distributed Systems

- E.g., a parallel file system, with 768 I/O nodes >50K disks
  - Systems are built to tolerate disk and node failures
  - However, most failures in the field are due to "performance bugs": e.g., time-outs, due to thrashing

- How do we build feedback mechanisms that ensure stability?
- How do we provide quality of service?
- What is a quantitative theory of resilience? (E.g. Impact of failure rate on overall performance)
  - Focus on systems where failures are not exceptional

Communication

- Communication-efficient algorithms
- A better understanding of fundamental communication-computation tradeoffs for PDE: tradeoffs between communication and convergence rate
- Programming models, libraries and languages where communication is a first-class citizen
  - Including shared memory models: (non-coherent) shared memory + explicit data movement

Asynchrony

- Asynchrony-tolerant algorithms
- What is a measure of asynchrony tolerance?
  - Moving away from the qualitative (e.g., wait-free) to the quantitative:
  - How much do intermittently slow processes slow down the entire computation – on average?
- What are the trade-offs between synchronicity and computation work?

Conclusion

- Moore's Law is slowing down; the slow-down has many fundamental consequences – only a few of them explored in this talk

- HPC is the "canary in the mine":
  - issues appear earlier because of size and tight coupling

- Optimistic view of the next decades: no stasis.
  - A frenzy of innovation to continue pushing current ecosystem, followed by frenzy of innovation to use totally different compute technologies

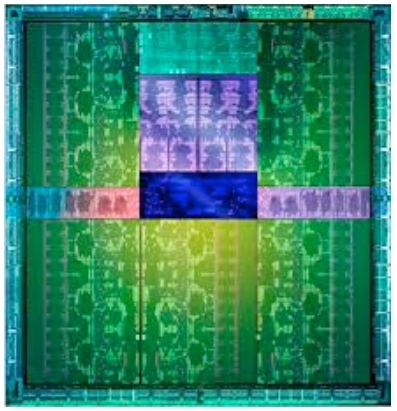
- Pessimistic view: The end is coming

Resilient Parallel Algorithms - Overcoming Silent Data Corruptions

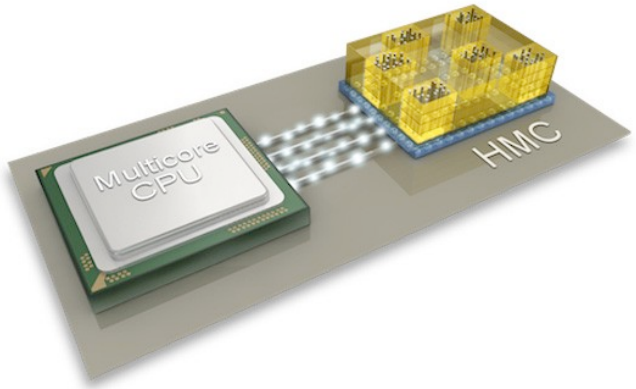
- SDCs may be unavoidable in future large systems (due to flips in computation logic)
- Intuition: SDC can either be
  - Type 1: Grossly violates the computation model (e.g. jump to wrong address, message sent to wrong node), or
  - Type 2: Introduces noise in the data (bit flip in a large array)
- Many iterative algorithms can tolerate infrequent type 2 errors
- Type 1 errors are often catastrophic and easy to detect in software

- Can we build systems that avoid or correct easy to detect (type 1) errors and tolerate hard to detect (type 2) errors?
- What is the general theory of fault-tolerant numerical algorithms?

Architecture-Specific Algorithms



- GPU/accelerators
- Hybrid memory Cube / Near-memory computing
- NVRAM – E.g., flash memory



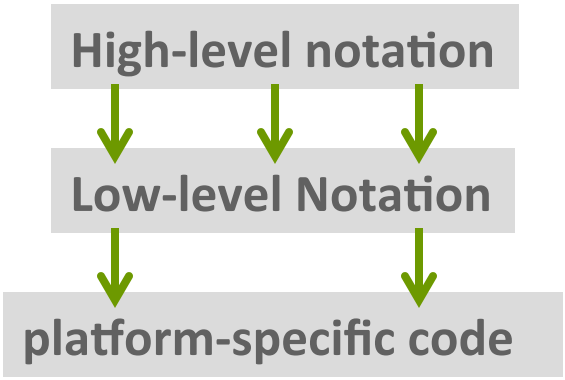
Portable Performance - High-Level Compilation

Practice

Code A → **Code B** → **Code C**

Manual conversion
"ifdef" spaghetti

Ideal



- Can we redefine "compilation" so that:
  - It starts at a higher level than now – includes code tuning
  - It supports well a human in the loop (manual high-level decisions vs. automated low-level transformations: "performance refactoring")
  - It integrates auto-tuning and profile-guided compilation
  - It preserves high-level code semantics
  - It preserves high-level code "performance semantics"

Exascale Design Point 202x with a cap of $200M and 20MW

Systems	2012 BG/Q Computer	2020-2024	Difference Today & 2019
System peak	**20 Pflop/s**	**1 Eflop/s**	**O(100)**
Power	**8.6 MW**	**~20 MW**	
System memory	1.6 PB (16*96*1024)	32 - 64 PB	O(10)
Node performance	205 GF/s (16*1.6GHz*8)	1.2 or 15TF/s	O(10) – O(100)
Node memory BW	42.6 GB/s	2 - 4TB/s	O(1000)
Node concurrency	64 Threads	O(1k) or 10k	O(100) – O(1000)
Total Node Interconnect BW	20 GB/s	200-400GB/s	O(10)
System size (nodes)	98,304 (96*1024)	O(100,000) or O(1M)	O(100) – O(1000)
Total concurrency	5.97 M	O(billion)	O(1,000)
MTTI	4 days	O(<1 day)	- O(10)

Both price and power envelopes may be too aggressive!

Moore's Law (2010)

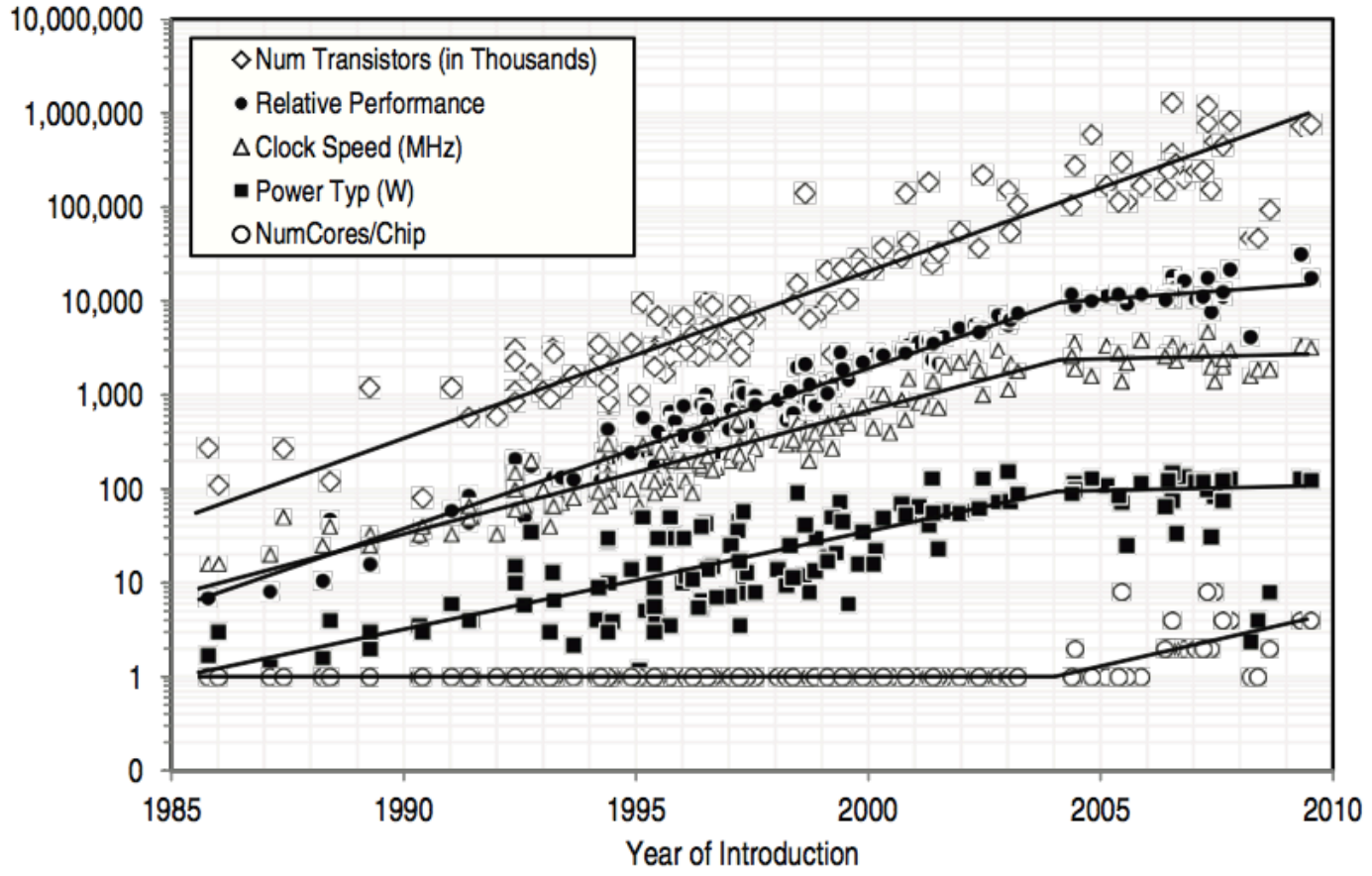
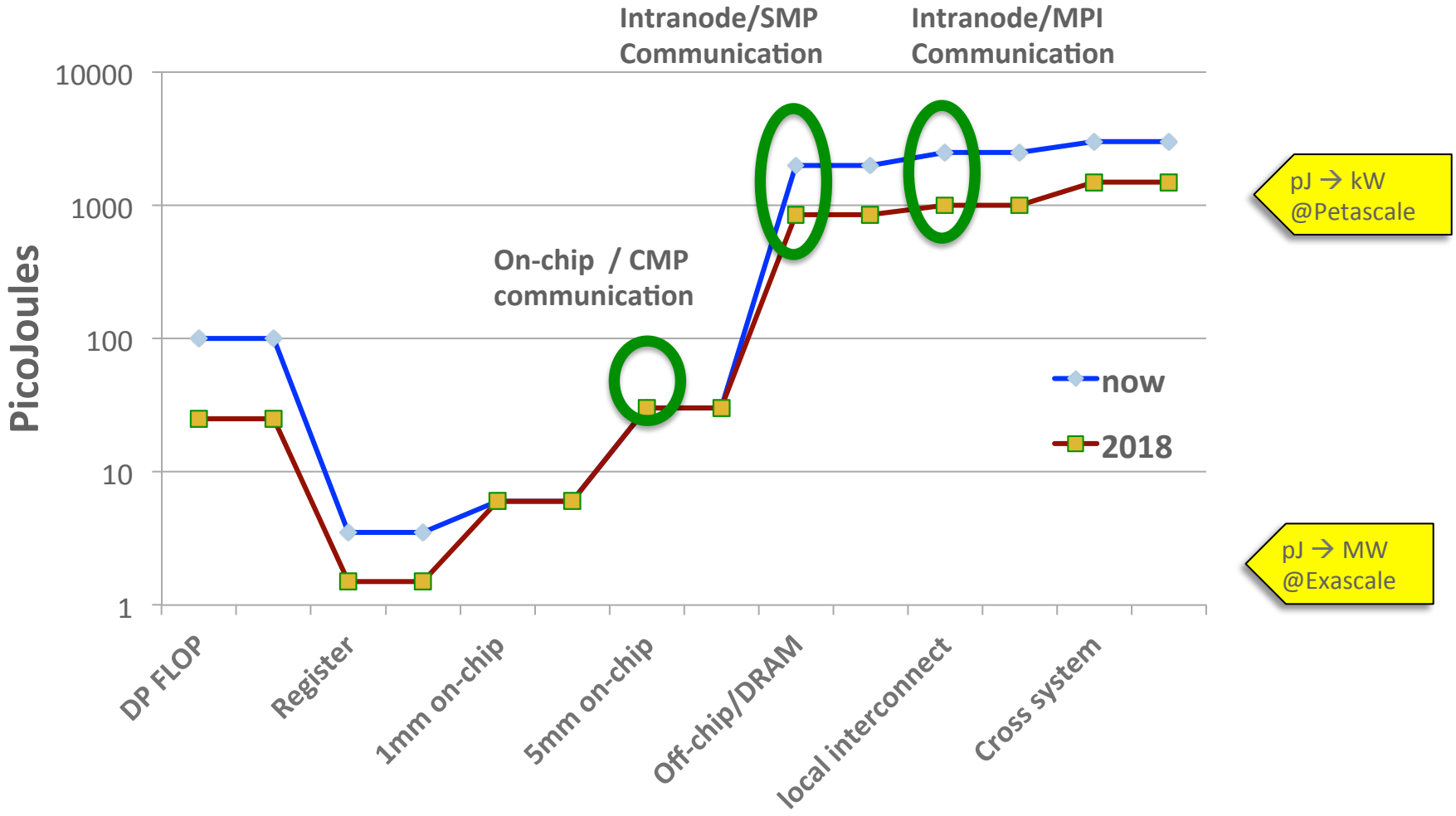


FIGURE 2.1 Transistors, frequency, power, performance, and cores over time (1985-2010). The vertical scale is logarithmic. Data curated by Mark Horowitz with input from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanoviç.

The Future of Computing Performance: Game Over or Next Level, Samuel Fuller and Lynette Millet, Eds., National Academy Press, 2011

Memory Bandwidth & Data Movement Performance will be Energy limited



ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems, Peter Kogge, Editor and Study Lead, DARPA-TR-2008-13, 2008.