UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

Supercomputing: Past, Present, Future

Marc Snir

illinois.edu

THE JURASSIC ERA OF SUPERCOMPUTING

illinois.edu

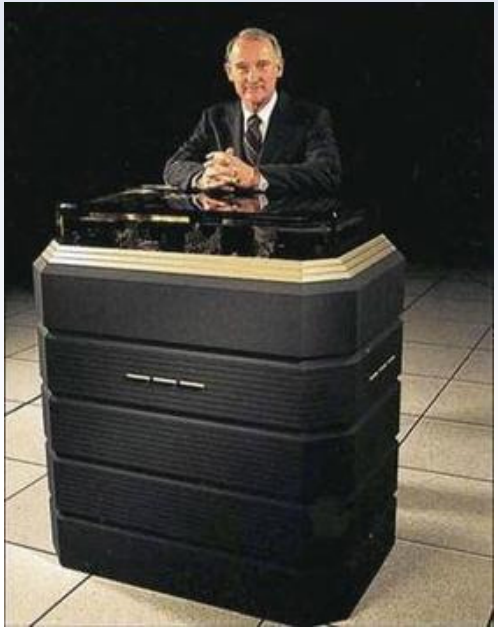
Those who cannot remember the past are condemned to repeat it. (Santayana)

Cray 1

- Vector supercomputer
- 1976
- 12.5 nsec clock
- 160-250 MFLOPs
- 1 Mword memory
- ~250,000 ECL gates
- Freon cooling
- 115 kW (not including cooling and I/O)

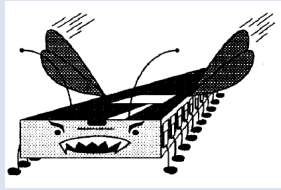
The End – Cray 3

- 1993
- GaAs gates
- 2.2 nsec
- ~1.9 GFLOPs
- up to 2 GW
- Immersive fluoerinert cooling
- 16 Gflops (?)
- Was never sold (Cray went bankrupt)



The 1990 Big Extinction: The Attack of the Killer Micros

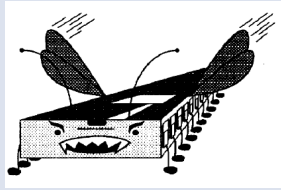
(Eugene Brooks, 1990)



Shift from bipolar vector machines & to clusters of MOS micros

The 1990 Big Extinction: The Attack of the Killer Micros

(Eugene Brooks, 1990)

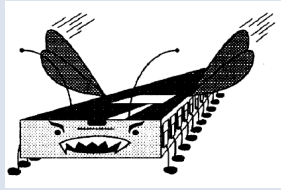


Shift from bipolar vector machines & to clusters of MOS micros

- *Roadblock*: bipolar circuits leaked too much current – it became too hard to cool them (even with liquid nitrogen)

The 1990 Big Extinction: The Attack of the Killer Micros

(Eugene Brooks, 1990)

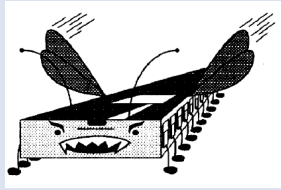


Shift from bipolar vector machines & to clusters of MOS micros

- *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

The 1990 Big Extinction: The Attack of the Killer Micros

(Eugene Brooks, 1990)

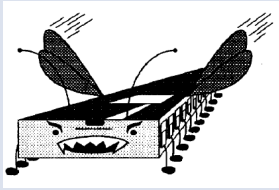


Shift from bipolar vector machines & to clusters of MOS micros

- *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

The 1990 Big Extinction: The Attack of the Killer Micros

(Eugene Brooks, 1990)

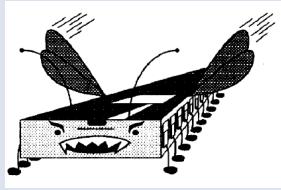


Shift from bipolar vector machines & to clusters of MOS micros

- *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")

The 1990 Big Extinction: The Attack of the Killer Micros

(Eugene Brooks, 1990)

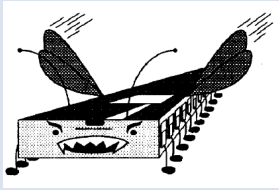


Shift from bipolar vector machines & to clusters of MOS micros

- *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 (good enough disruptive technology – Christensen)

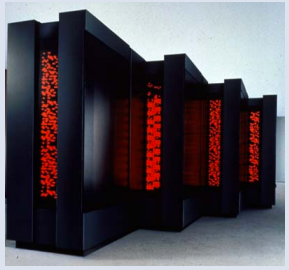
The 1990 Big Extinction: The Attack of the Killer Micros

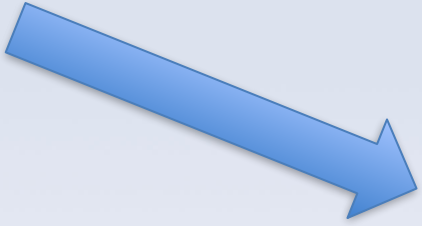
(Eugene Brooks, 1990)

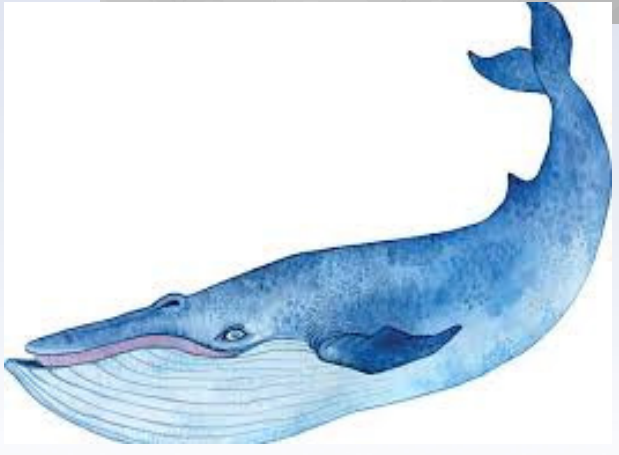


Shift from bipolar vector machines & to clusters of MOS micros

- *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 (good enough disruptive technology – Christensen)
  – Cray C90 vs. CM5 in 1991: 244 MHz vs. 32 MHz







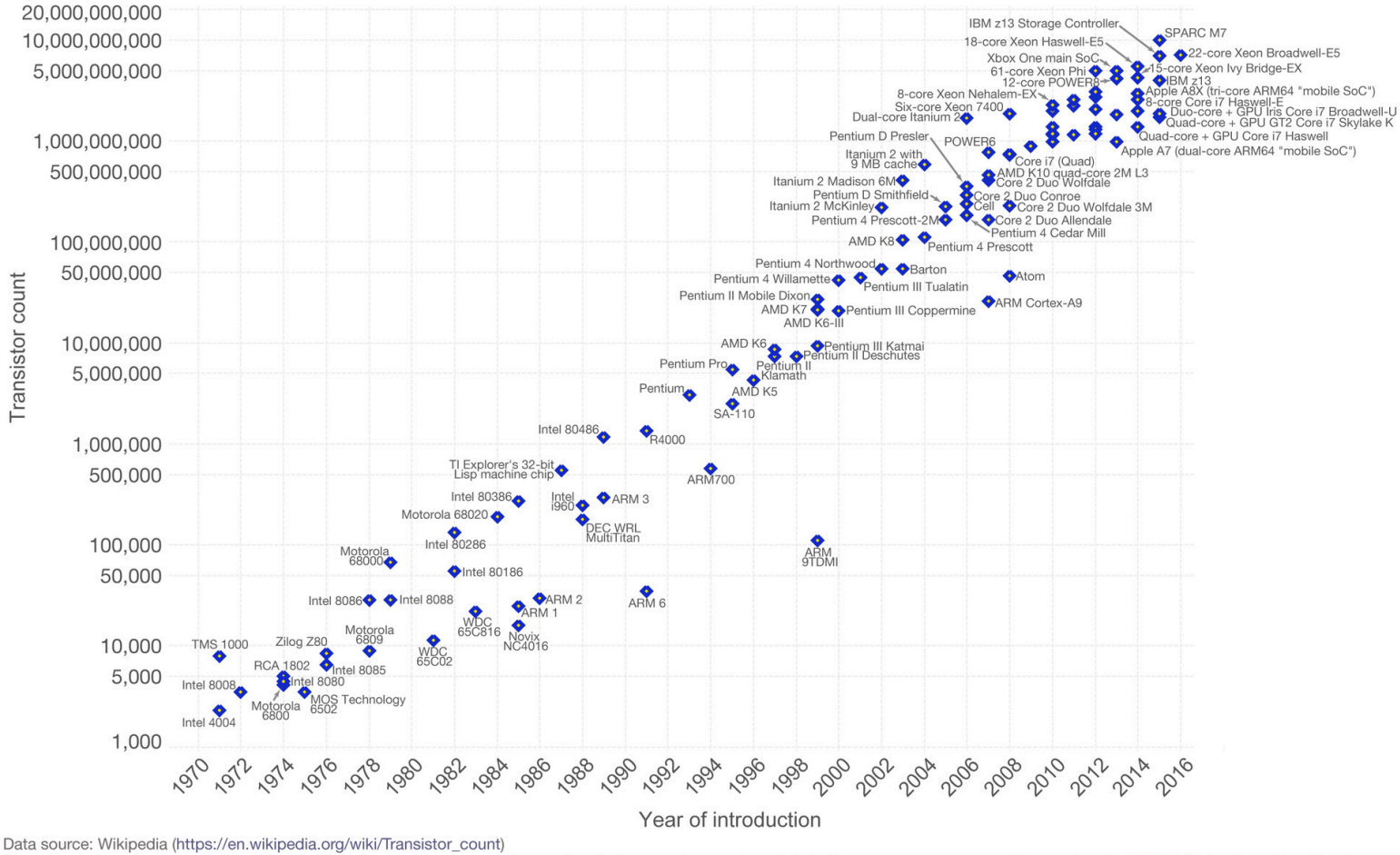
THE AGE OF THE MAMMALS: MOORE'S LAW

#transistors per chip doubling every two years

Moore's Law – The number of transistors on integrated circuit chips (1971-2016)

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.



Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

25 Years = 10^6: from 200 Gflops to 200 Pflops

25 Years = 10^6: from 200 Gflops to 200 Pflops

1. Ride Moore's Law – use chips with more and more transistors (~x10^4)

25 Years = 10^6: from 200 Gflops to 200 Pflops

1. Ride Moore's Law – use chips with more and more transistors (~x10^4)
2. Run them faster (~x10^2)
   – that stopped > 10 years ago

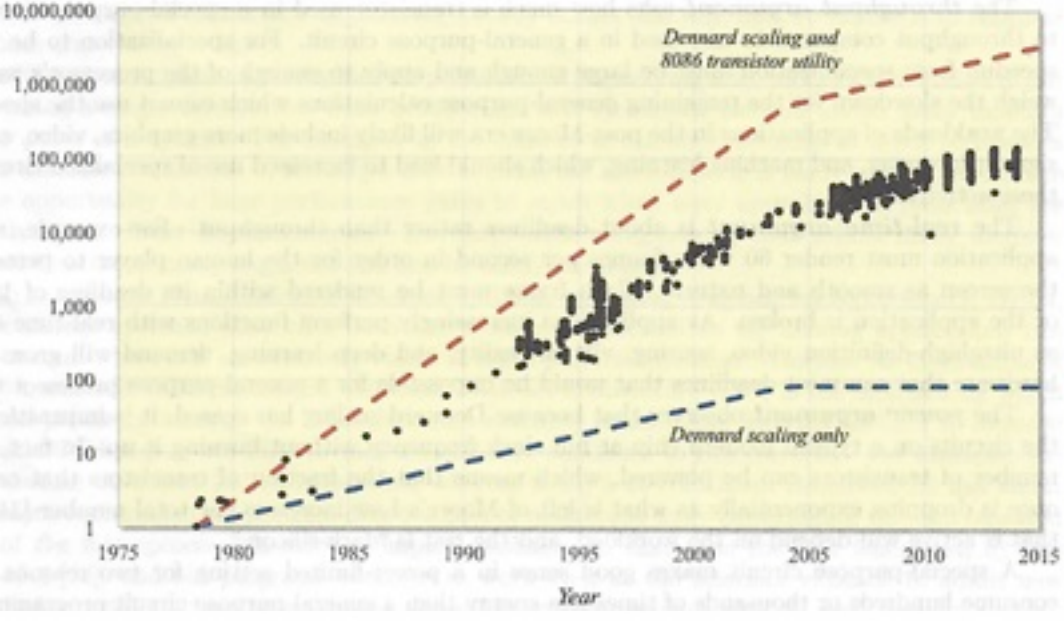
25 Years = 10^6: from 200 Gflops to 200 Pflops

1. Ride Moore's Law – use chips with more and more transistors (~x10^4)
2. Run them faster (~x10^2)
   – that stopped > 10 years ago
3. Build bigger (and more expensive) machines (~x10)

25 Years = 10^6: from 200 Gflops to 200 Pflops

1. Ride Moore's Law – use chips with more and more transistors (~x10^4)
2. Run them faster (~x10^2)
   – that stopped > 10 years ago
3. Build bigger (and more expensive) machines (~x10)
4. Use mass-produced components wherever possible
   – So as to be on the same cost-performance curve as mass-produced devices and software
   • Over the years, customization has decreased! (no specialized OS, no specialized networks…); trend continues, with cloud technologies
   – *Efficiency has decreased!*

Loss of Efficiency -- SPECint



(Leiserson et al, There's Plenty of Room at the Top)

Theoretical gain if efficiency were preserved

Actual gain (benchmarks)

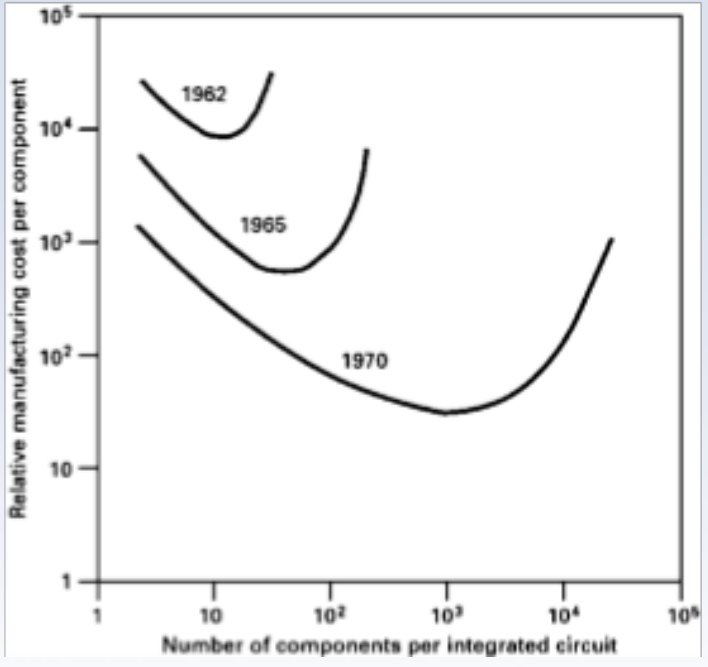
Gain from faster clock

Moore's Law

- Not a law of nature, but of economics:

The cost per transistor is minimized by doubling the number of transistors per chip every two years.

- Achieved by technological progress and by increasing manufacturing volumes
  - Due to growth in market size AND increased industry consolidation
  - 20 companies produced 130 nm chips; only 4 produce ~10 nm chips: TSMC, Global Foundries, Samsung and Intel



Proceedings of the IEEE | Vol. 103, No. 10, October 2015

Moore’s Law is Dying

Moore’s Law is Dying

- Technological wall reached at ~5nm (number of atoms per gate too small)

Moore’s Law is Dying

- Technological wall reached at ~5nm (number of atoms per gate too small)
- Continued evolution is increasingly expensive (both non-recurring costs and manufacturing costs)
  - Consolidation is not an option anymore

Moore’s Law is Dying

- Technological wall reached at ~5nm (number of atoms per gate too small)
- Continued evolution is increasingly expensive (both non-recurring costs and manufacturing costs)
  - Consolidation is not an option anymore
- Performance gains from continued miniaturization are decreasing

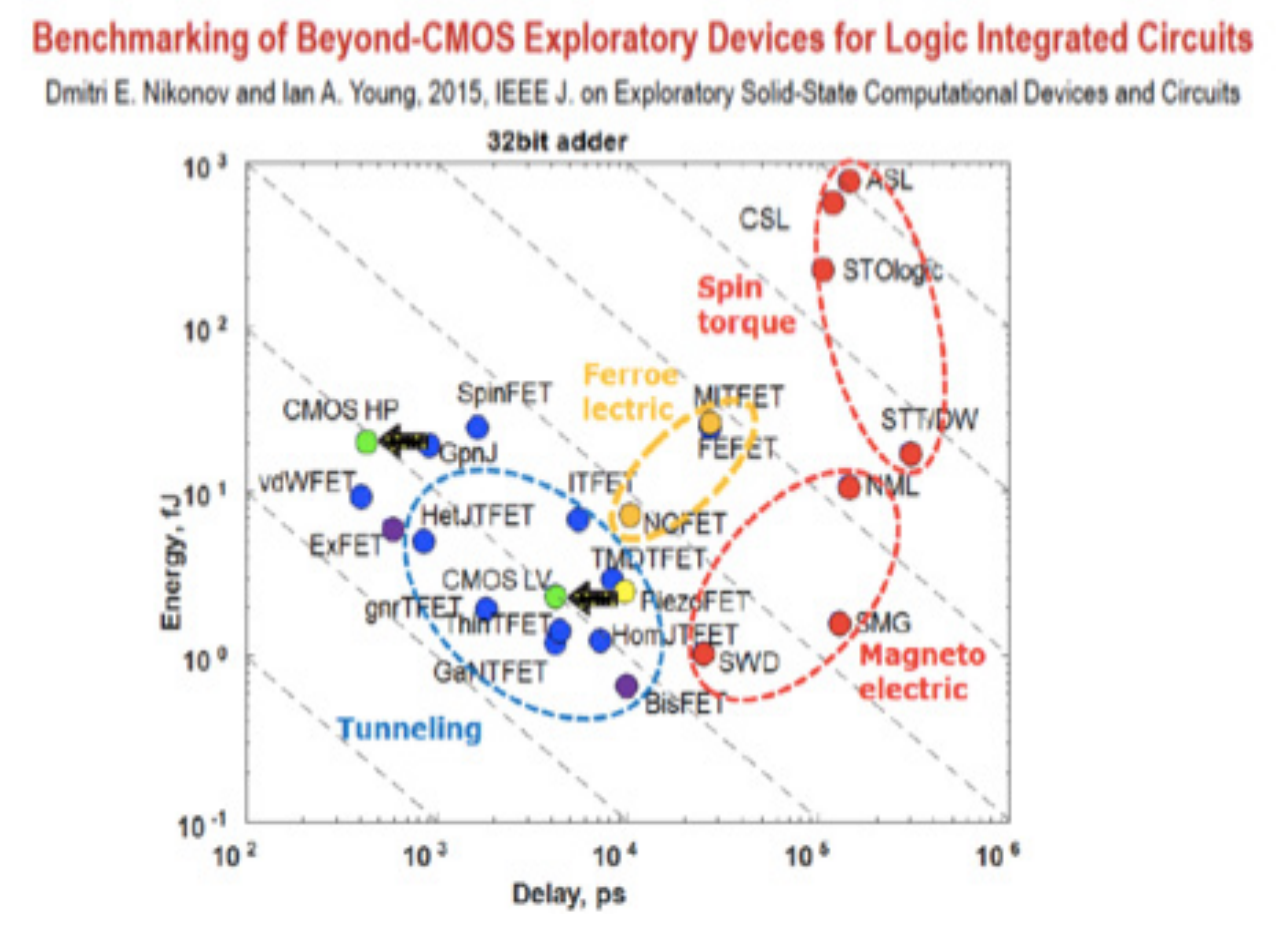
Moore's Law is Dying

- Technological wall reached at ~5nm (number of atoms per gate too small)
- Continued evolution is increasingly expensive (both non-recurring costs and manufacturing costs)
  - Consolidation is not an option anymore
- Performance gains from continued miniaturization are decreasing
- Deployment of new silicon generations has slowed down

WHAT NEXT?

Evolution: Faster and/or Lower Power Gates

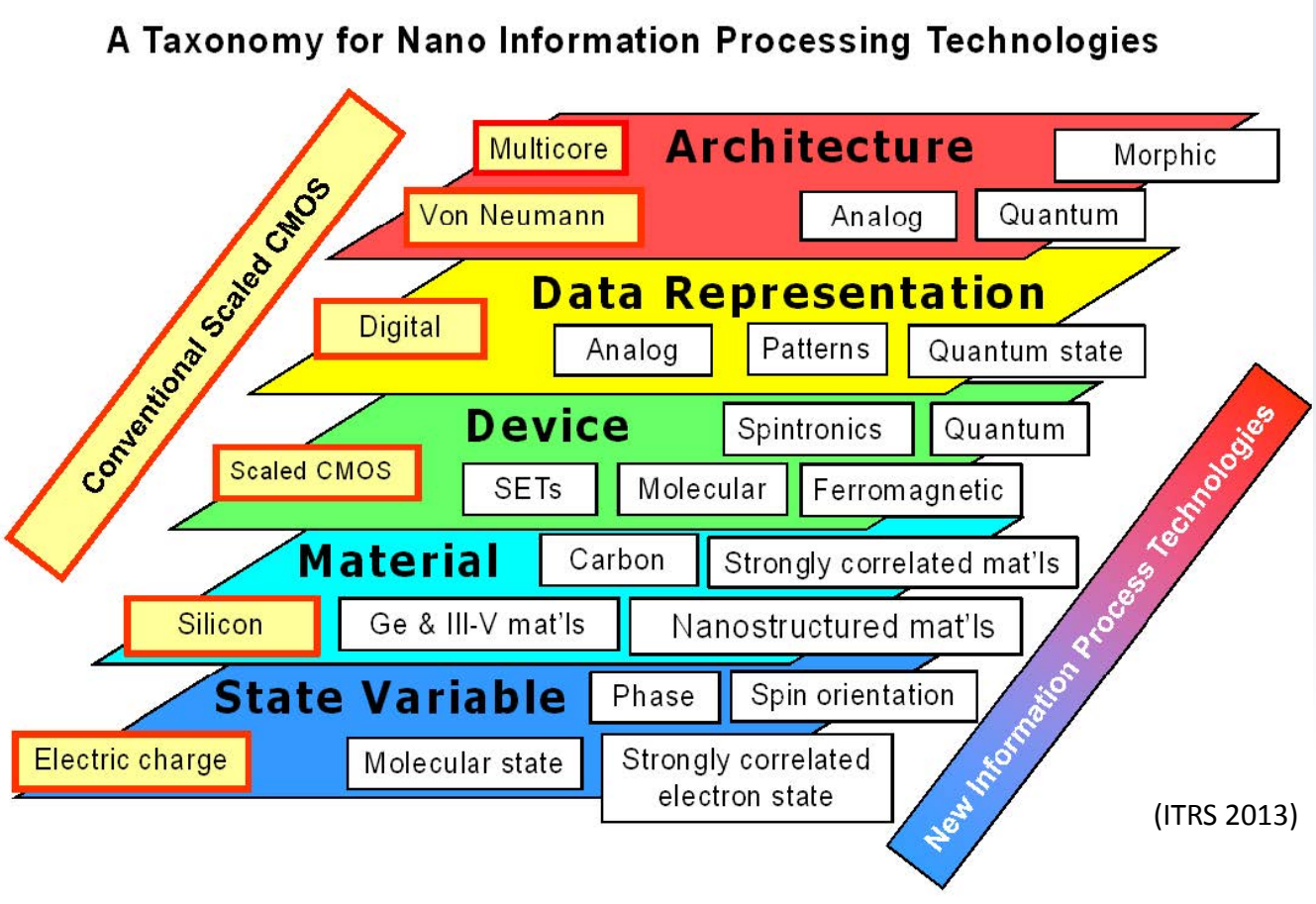
- Goal: Lower speed×energy product
- No technology offers significant advantage over CMOS
- ~10 year gap from device in lab to manufactured chip



illinois.edu

Revolution: Totally Different Technology

- No technology is ready for prime time; many will fai
- Many (e.g., quantum, analog optics) are not general-purpose
- Need to exploit opportunities above device level

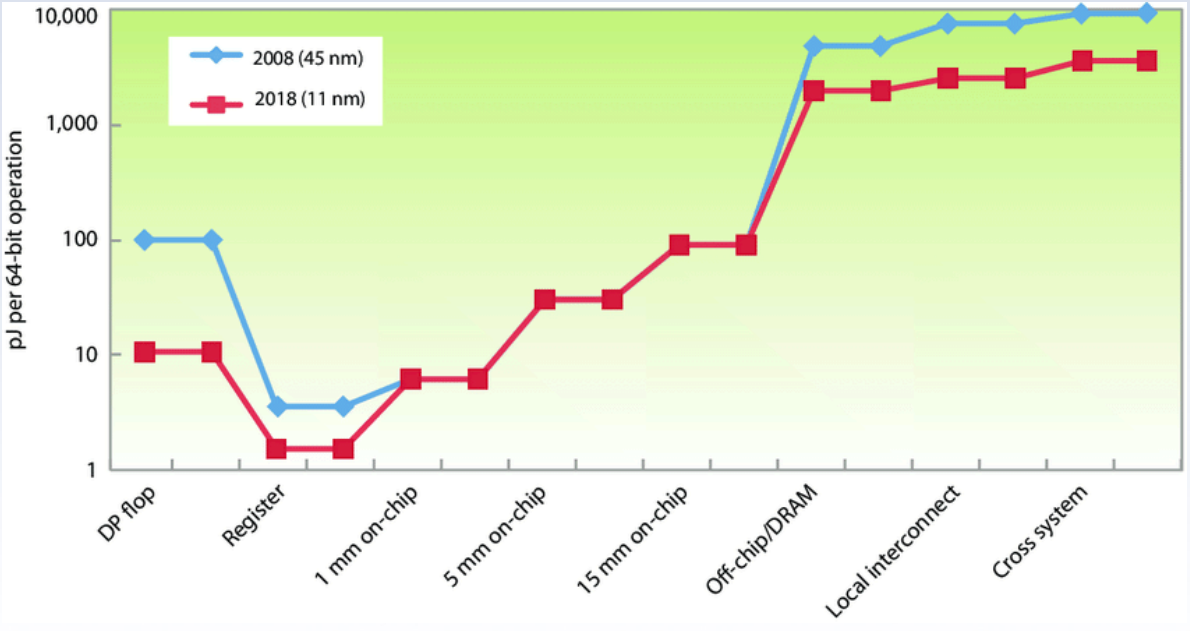


(ITRS 2013)

PACKAGING & COOLING

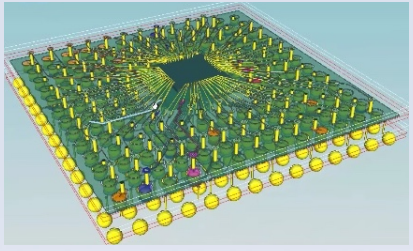
Increase Density and Improve Interchip Communication

Crossing chip boundaries is slow and energy-expensive

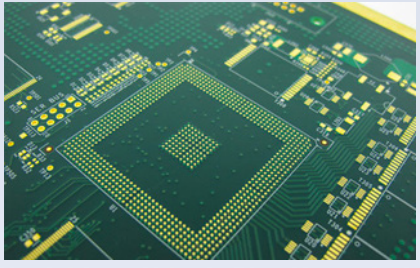


(Kogge & Shalf, Exascale Computing Trends: Adjusting to the "New Normal"' for Computer Architecture Computing in Science and Engineering Nov 2013)

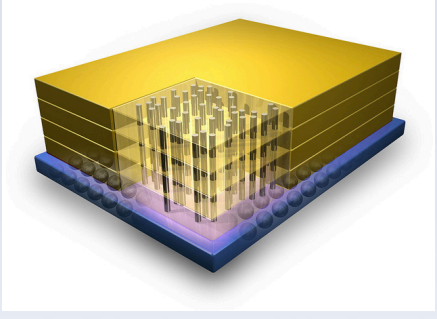
Technologies



chip package

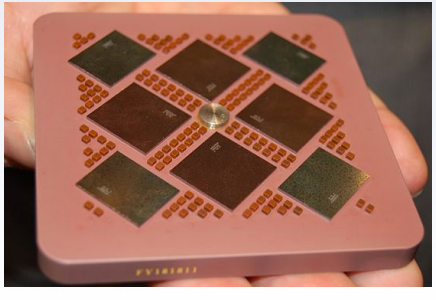


printed circuit board



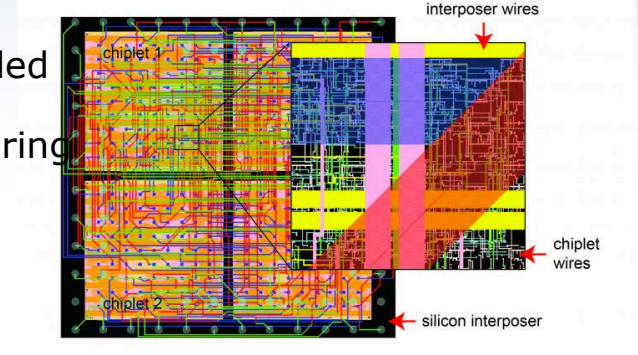
vertical packaging

multi chip module



- Can build larger tightly-coupled system
- Lower design and manufacturing costs (better yield)
- Easier customization

chiplets on silicon interposer



(GaTech)

ARCHITECTURE

Processor Specialization & Adaptation

Processor Specialization & Adaptation

- Dynamically reconfigurable hardware (heterogeneity in time)
  - We control today frequency at chip level
  - Will control frequency/power for individual components (cores, memory controllers) and will be able to gate them
  - May use FPGAs

Processor Specialization & Adaptation

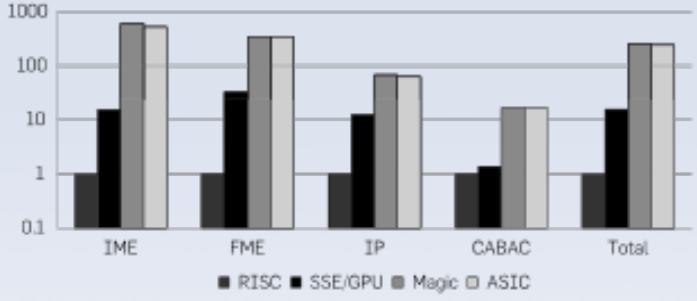
- Dynamically reconfigurable hardware (heterogeneity in time)
  - We control today frequency at chip level
  - Will control frequency/power for individual components (cores, memory controllers) and will be able to gate them
  - May use FPGAs
- Specialized accelerators (heterogeneity in space)

Processor Specialization & Adaptation

- Dynamically reconfigurable hardware (heterogeneity in time)
  - We control today frequency at chip level
  - Will control frequency/power for individual components (cores, memory controllers) and will be able to gate them
  - May use FPGAs
- Specialized accelerators (heterogeneity in space)
- Specialized memories (ibid)

(Hameed et al, Understanding Sources of Inefficiency in General-Purpose Chips. CACM 1Oct 2011)

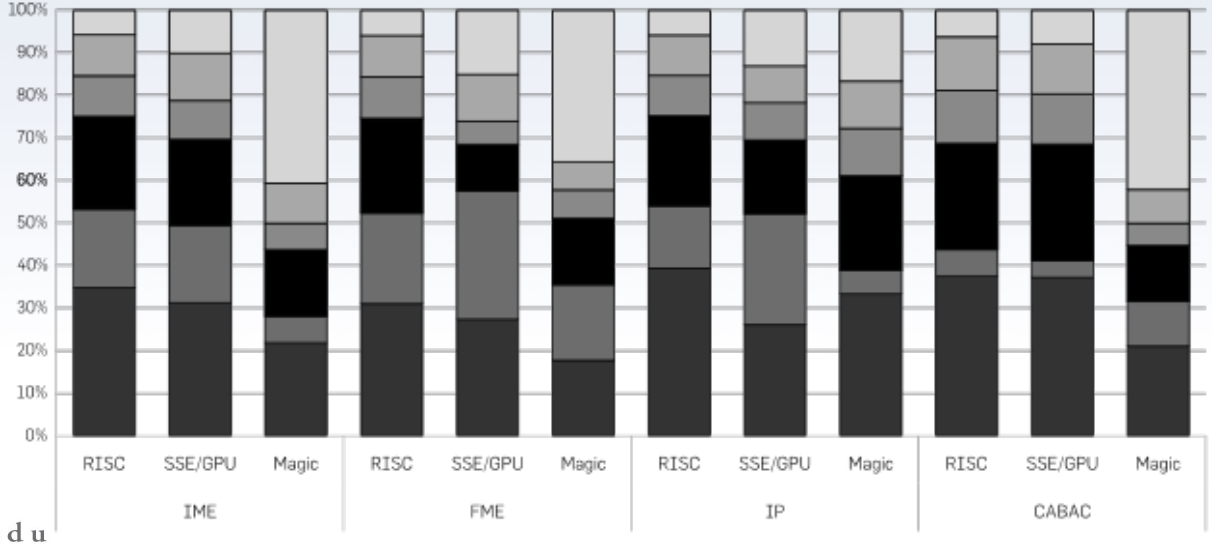
ASIC vs. Software (H.264 encoder)



speedup

- Main contribution of ASIC is to reduce overheads

energy



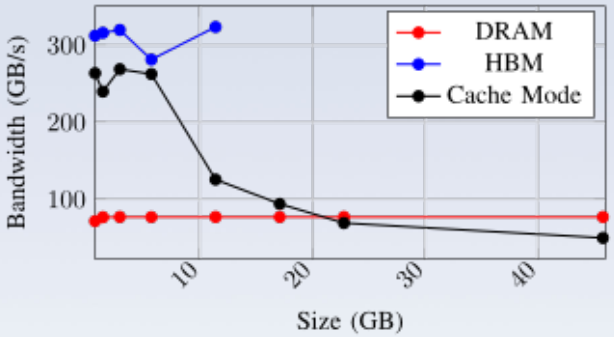
- functional units
- register files
- control
- pipeline registers, buses
- data cache
- instruction fetch/ decode

illinois.edu

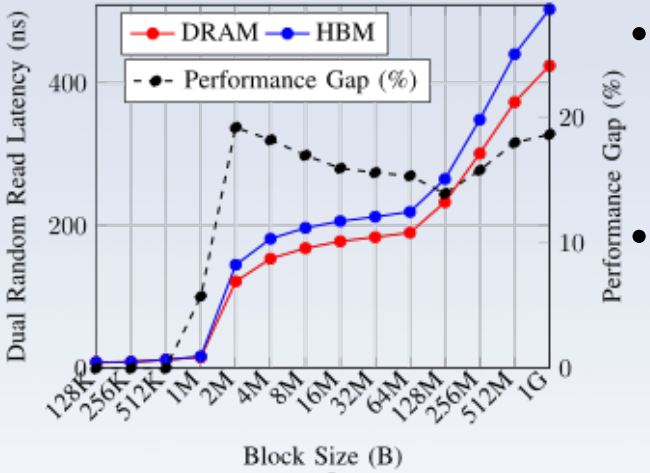
(Peng, et al, Exploring the Performance Benefit of Hybrid Memory System on HPC Environments. IPDPSW 2017)

KNL: DDR4 vs. MCDRAM (HBM)

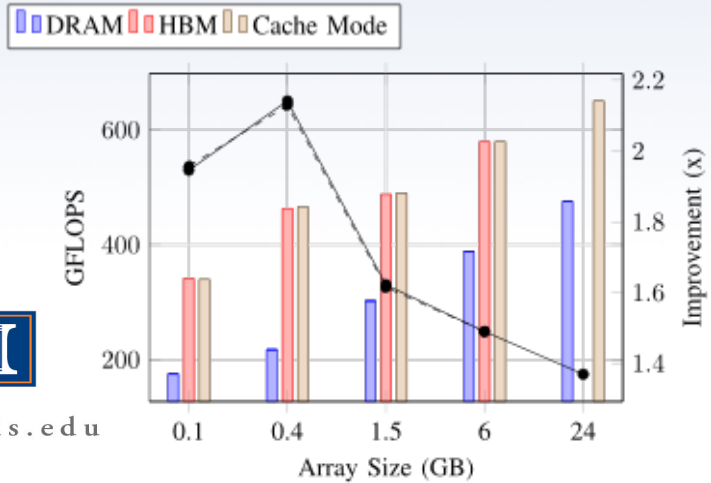
Stream

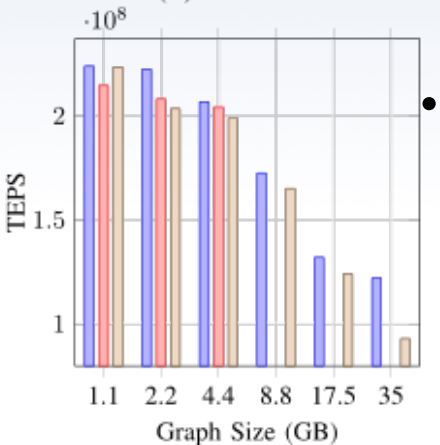


Random access



DGEMM





- MCDRAM has x4 bandwidth of DDR
- MCDRAM is limited to 16 GB while DDR can grow to 384 GB
- MCDRAM has higher latency

Graph500

illinois.edu

Remarks (1)

Remarks (1)

- Specialized hardware has less of a handicap with the end of Moore's Law
  - One design has longer lifetime
  - Can use "dark silicon"

Remarks (1)

- Specialized hardware has less of a handicap with the end of Moore's Law
  - One design has longer lifetime
  - Can use "dark silicon"
- ASIC – Application-Specific Integrated Circuit, is not an option for HPC (too many apps); we can use algorithm specific or computation-pattern specific chip
  - E.g., FFT or Molecular Dynamic accelerators

Remarks (1)

- Specialized hardware has less of a handicap with the end of Moore's Law
  - One design has longer lifetime
  - Can use "dark silicon"
- ASIC – Application-Specific Integrated Circuit, is not an option for HPC (too many apps); we can use algorithm specific or computation-pattern specific chip
  - E.g., FFT or Molecular Dynamic accelerators
- How many distinct accelerators are needed to cover a high fraction of scientific computing cycles?

Issues (2)

Issues (2)

- Hardware components are not all equal and change over time (power management, fault tolerance)

Issues (2)

- Hardware components are not all equal and change over time (power management, fault tolerance)
- Applications are increasingly dynamic and irregular

Issues (2)

- Hardware components are not all equal and change over time (power management, fault tolerance)
- Applications are increasingly dynamic and irregular
- How do you map an evolving computation onto a heterogeneous, evolving platform?

Issues (2)

- Hardware components are not all equal and change over time (power management, fault tolerance)
- Applications are increasingly dynamic and irregular
- How do you map an evolving computation onto a heterogeneous, evolving platform?

- Current trend:
  - Higher level, "hybrid dataflow" type code
    - codelets and dependencies
    - with data reuse – hence can express locality
  - Intelligent runtime that maps codelets to resources and event-driven scheduler

Issues (2)

- Hardware components are not all equal and change over time (power management, fault tolerance)
- Applications are increasingly dynamic and irregular
- How do you map an evolving computation onto a heterogeneous, evolving platform?

- Current trend:
  - Higher level, "hybrid dataflow" type code
    - codelets and dependencies
    - with data reuse – hence can express locality
  - Intelligent runtime that maps codelets to resources and event-driven scheduler
- Legion (Aiken), Parsec (Bosilca)…

Issues (2)

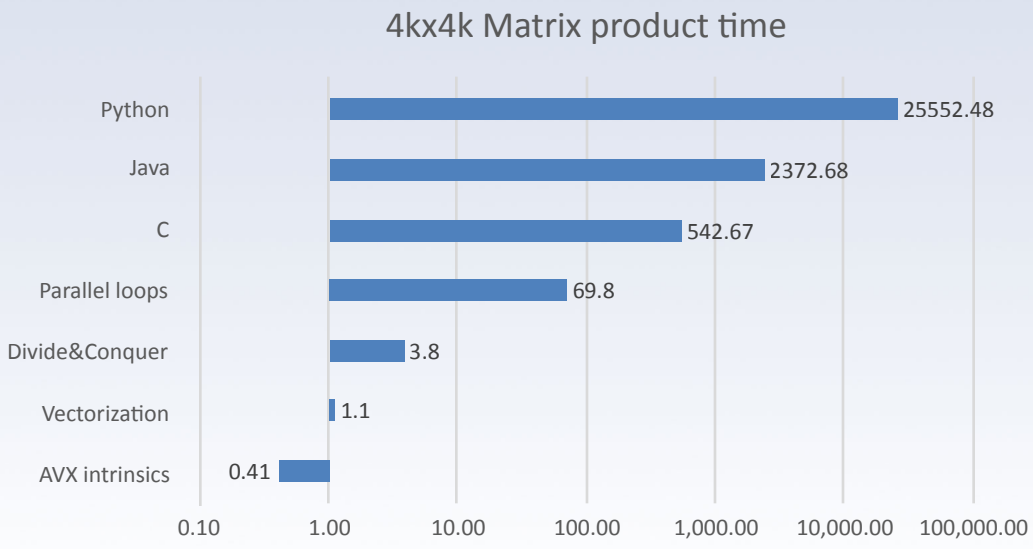
- Hardware components are not all equal and change over time (power management, fault tolerance)
- Applications are increasingly dynamic and irregular
- How do you map an evolving computation onto a heterogeneous, evolving platform?

- Current trend:
  - Higher level, "hybrid dataflow" type code
    - codelets and dependencies
    - with data reuse – hence can express locality
  - Intelligent runtime that maps codelets to resources and event-driven scheduler
- Legion (Aiken), Parsec (Bosilca)…
- Need more asynchronous algorithms and better support for producer-consumer synchronization in hardware and firmware

COMPILER &RUNTIME

Issue

- Compilers do not properly map code to hardware, even in the simplest case



(Leiserson et al, There's Plenty of Room at the Top)

Possible Solutions

Possible Solutions

- “Elbow grease” (hard work)

Possible Solutions

- “Elbow grease” (hard work)
- Autotuning: search for optimal code configuration

Possible Solutions

- “Elbow grease” (hard work)
- Autotuning: search for optimal code configuration
- Interactive autotuning?

Possible Solutions

- “Elbow grease” (hard work)
- Autotuning: search for optimal code configuration
- Interactive autotuning?
- Application of ML?

Possible Solutions

- “Elbow grease” (hard work)
- Autotuning: search for optimal code configuration
- Interactive autotuning?
- Application of ML?

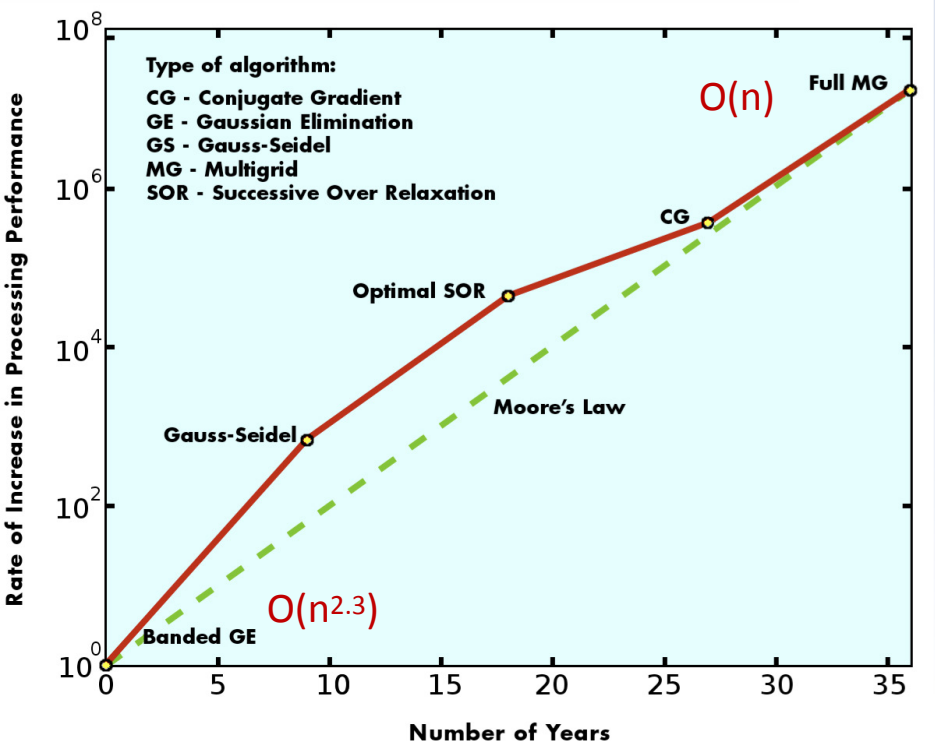
Possible Solutions

- "Elbow grease" (hard work)
- Autotuning: search for optimal code configuration
- Interactive autotuning?
- Application of ML?

- Need to collect training data in a much more extensive and systematic manner!

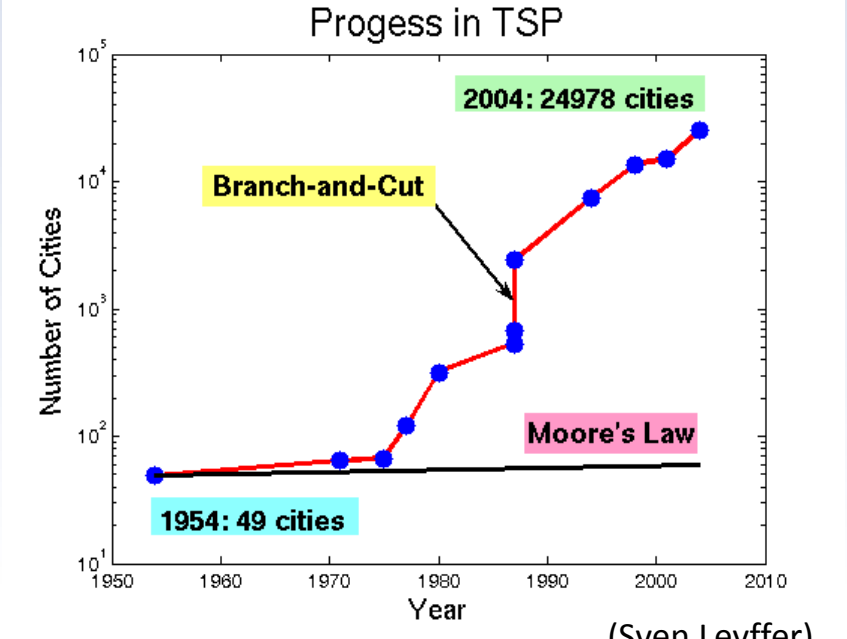
ALGORITHMS

Algorithmic Improvements Often Exceed Moore's Law Contribution



Poisson solver n=64x64x64

(A SCIENCE-BASED CASE FOR LARGE-SCALE SIMULATION DOE SC 2003)



(Sven Leyffer)

Traveling Salesman Problem

Is There Still Room for Improvements?

Is There Still Room for Improvements?

- A lot, for exponential optimization problems (such as TSP)

Is There Still Room for Improvements?

- A lot, for exponential optimization problems (such as TSP)
- A lot, for aleatoric systems (estimate probability distribution – infinite dimensional space)

Is There Still Room for Improvements?

- A lot, for exponential optimization problems (such as TSP)
- A lot, for aleatoric systems (estimate probability distribution – infinite dimensional space)
- Much, even for problems where we seem to be at the end of the road – as we got to linear complexity

What is Problem Size n?

What is Problem Size n?

- Consider a PDE iterative solver
  - Can use meshes of different resolutions
  - Can use adaptive meshes
  - Can use lower precision arithmetic
  - Can compress data
  - Can allow for occasional bit flips
  - Can leverage knowledge of what the output is used for

What is Problem Size n?

- Consider a PDE iterative solver
  - Can use meshes of different resolutions
  - Can use adaptive meshes
  - Can use lower precision arithmetic
  - Can compress data
  - Can allow for occasional bit flips
  - Can leverage knowledge of what the output is used for
- Can sample, for aleatoric problems

What is Problem Size n?

- Consider a PDE iterative solver
  - Can use meshes of different resolutions
  - Can use adaptive meshes
  - Can use lower precision arithmetic
  - Can compress data
  - Can allow for occasional bit flips
  - Can leverage knowledge of what the output is used for
- Can sample, for aleatoric problems

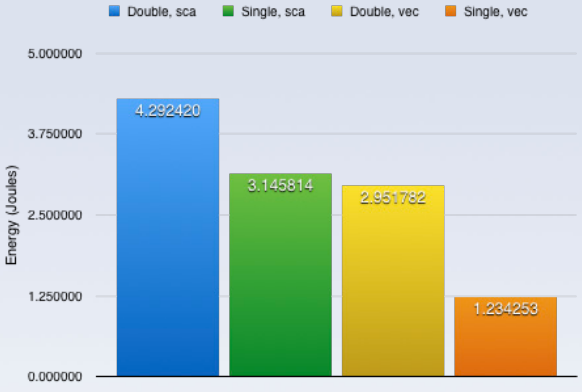
- How many bits are needed to solve a particular problem, and how stable these bits need be?

Lower Precision Arithmetic

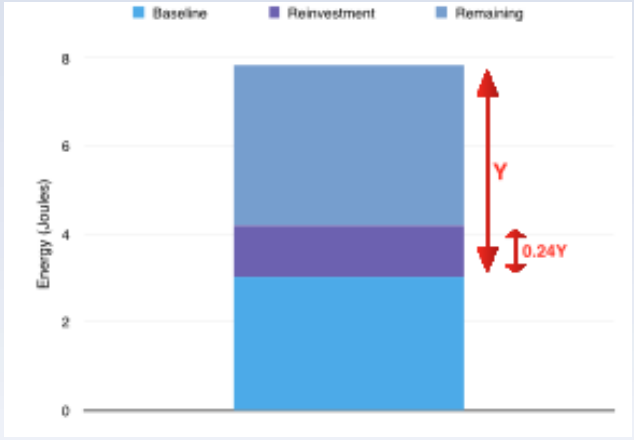
- Can run at least twice as fast with 32 bit arithmetic rather than 64 bits
  - Twice as many floating point operations, when using vector operations
  - Twice as many operands per memory access
  - Twice cache capacity
  - ~1/3 energy consumption

(Leyffer et a. lDoing Moore with Less--Leapfrogging Moore's Law with Inexactness for Supercomputing. arXiv Oct 2016)

Use Lower Precision



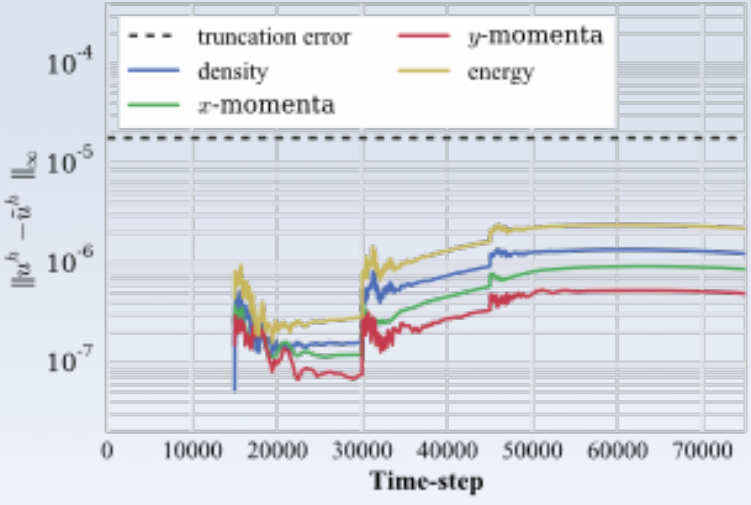
Energy consumed by inexact Newton solver of Rosenbrock Equation ($\varepsilon = 10^{-5}$)

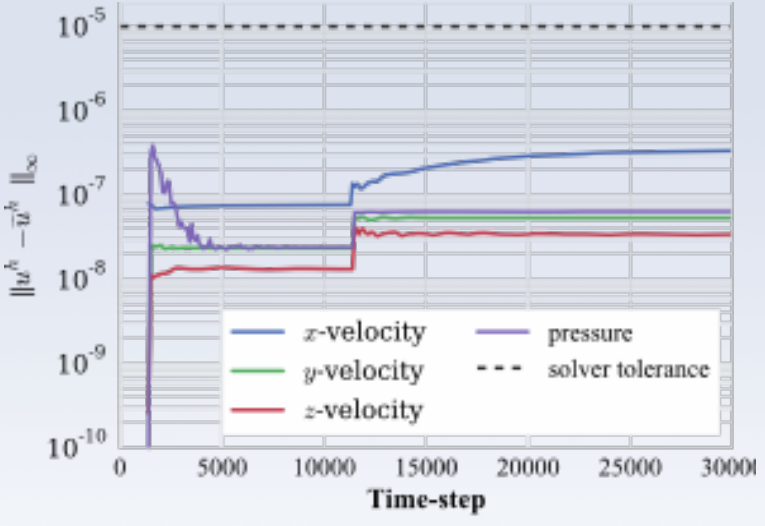


Run single-precision, next double precision ($\varepsilon = 10^{-13}$)

Reinvest energy saved by using lower precision in order to reduce error

Error Introduced by Lossy Compression of Checkpoint





PlasComCM

- Compression tolerance of ε=10^{-6}
- Compression ratio of ~x7

Nek5000

- Compression tolerance of ε=10^{-7}
- Compression ratio of ~x3

J. Calhoun et al Exploring the feasibility of lossy compression for PDE simulations IJHPCA, 2018

Can Cope with Occasional Bit-Flips

- Algorithm Based Fault Tolerance (ABFT): Can build algorithm specific fault detectors, using properties of the algorithm
- ML based fault detection: Can learn to detect error patterns for a specific solver (anomaly detection)

- Hypothesis: Errors either do not affect too much final solution or are easy to detect.

CONCLUSION

Future

- The end of Moore's Law is not the end of supercomputing
- Plenty of important problems can benefit from continued increase in performance
- Increasing performance will require increasing specialization, and moving away from commodity technologies
  - Different, specialized hardware
  - Specialized software
  - Focus on performance, not ease of programming
- Specialization is affordable when device technology is stable and justifiable due to the benefits from improved performance
- Think of a supercomputer not as a computer made large, but as a unique, expensive scientific instrument that cost billions, is used over decades, and require unique skills in order to use efficiently


Questions?