Computation matters less than communication and memory

Greg Astfalk

International Computing for the Atmospheric Sciences Symposium (iCAS2015)

September 15, 2015

Outline

- In this talk (I have 30 minutes, but need 3 hours) we'll quickly cover:
  - Performance shortcomings
  - Communication problems
  - Memory problems

Take-away “sound bites” for this talk

- In most high-end applications today we do not have a computation problem, we have a memory problem, and/or a communication problem
- The industry has spent years and many dollars engineering solutions to the wrong problem, that of increasing peak FLOPS
- Going forward we need to shift to focus on, and solve, the more critical problems; communications and memory

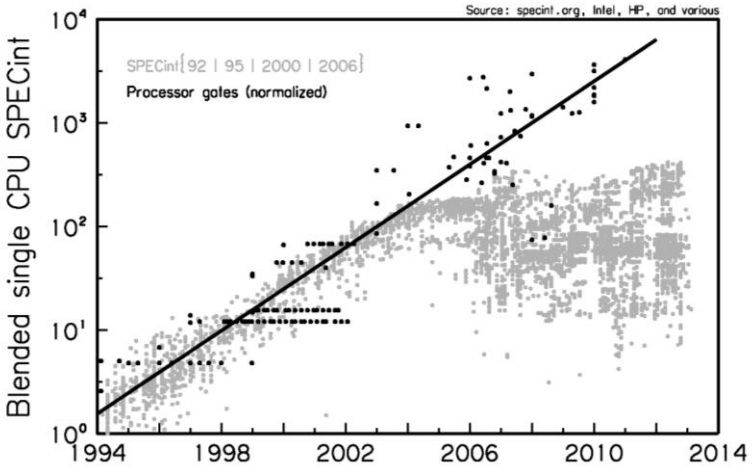
Computation

Performance

- A well known, seldom discussed, topic is application delivered performance as a percentage of peak
  - Single node 8–20%
  - Multi node 2–10%
- What we have done over history to solve this has been to increase the peak {FLOPS | OPS}
  - This has not helped (much)
- What we need are systems which have better performance/memory balance
  - More on a later slide

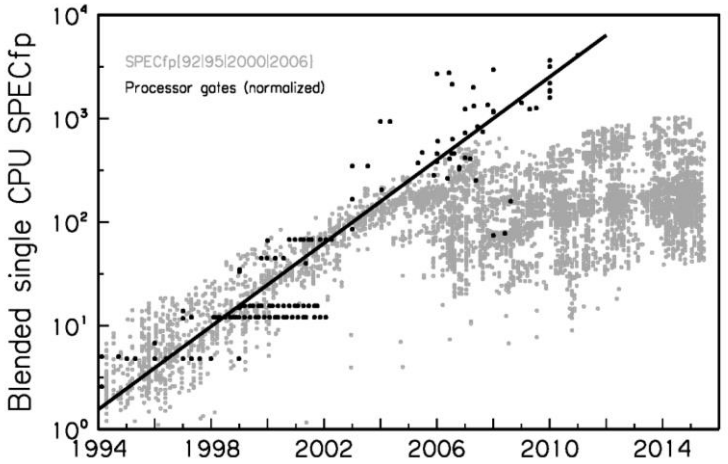
Performance vs. Moore's Law (1 of 2)

- No universal metric for measuring computer performance
- SPECint benchmark is a good proxy for a variety of applications
- Since 2003 the increasing number of transistors are not providing commensurate performance benefit



Performance vs. Moore's Law (2 of 2)

- If your interest is in floating-point performance, consider the SPECfp benchmark
  - Same issue
  - More performance potential that we don't get



Application signatures (your mileage may vary)

- For tuned “physics” codes
  - Only about 30% of instructions are floating-point
  - Majority of “vector” instructions are of length 1
  - Branches, “other”, and memory operations are the majority

Code performance

- For the WRF code†
  - The 12km CONUS benchmark from 2010 to 2014
    - Performance advanced from 22 to 44 Gflops
    - Percent of peak declined from 16% to 4%
  - Memory bandwidth is at peak for 13% of execution

† My thanks to Michael Lough of HP for his expertise with WRF

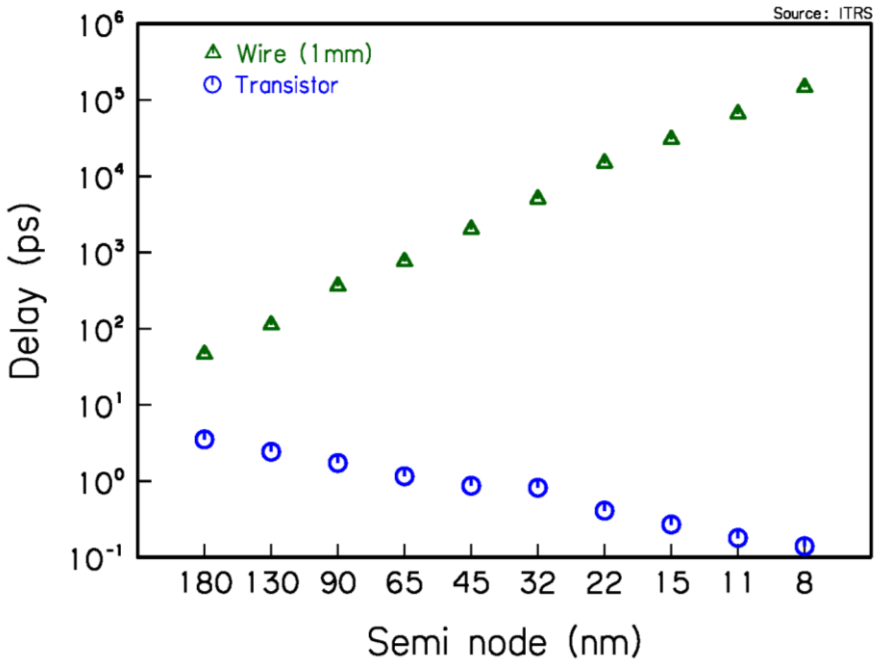
Communication

Data communications characteristics

- The key characteristics we are concerned with are:
  - Latency
  - Bandwidth
  - Energy per bit
  - “Reach”
- Each of these needs improvement

Data communications characteristics

- Transistors are getting faster
- Communication is getting slower



Communication energies

- Caveat: This data is FAR more complex than it may seem
  - Each of the 4 data cells requires a long explanation/discussion
  - The data are considering what it will be like 4–5 years from now

Energies, pJoules		
Reach	Electronics	Photonics
10 centimeters	25	~3
100 meters	$\mathcal{O}(10^4)$	$\mathcal{O}(10^1)$

Compute and communication energies

- More energy to move data than to compute on it
  - Computation almost feels “free” relative to communication
  - Time will make this worse

Operation	Energy (pJ)
64-bit integer operation	1
64-bit floating-point operation	20
256 bit on-die SRAM access	50
256 bit bus transfer (short)	26
256 bit bus transfer (1/2 die)	256
Off-die link (efficient)	500
256 bit bus transfer(across die)	1,000
DRAM read/write (512 bits)	16,000
HDD read/write	$\mathcal{O}(10^6)$

28nm CMOS, DDR3

What's needed

- Use silicon nanophotonics rather than copper and electrons
  - CMOS scale
  - CMOS cost curve
  - DWDM
  - Greater bits per unit time per unit of cross-sectional area or die edge
  - Distance invariance and reduced energy per bit
- Continue R&D to reduce the threshold length-scale for photonics to $\mathcal{O}(1)$ millimeter
  - We're almost there

Memory

The issue

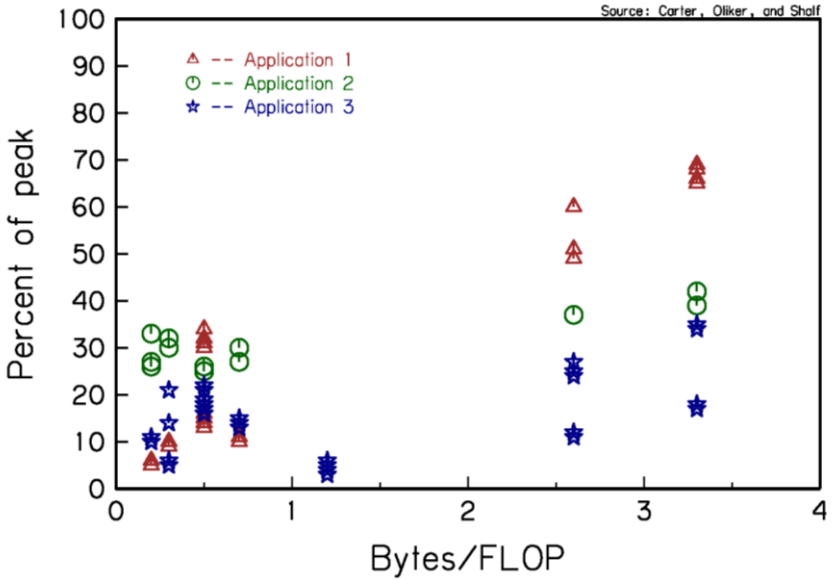
- Memory is the biggest performance issue today
- Why?
  - Has fallen way behind processors wrt bandwidth and capacity
  - It is not rapidly innovating
  - Must change the entire memory subsystem all at once
- Good news
  - We're getting close to fixing it

Old news

- My advocating for memory is not new
  - "…the real difficulty, the main bottleneck, of an automatic very high speed computing device lies: At the memory…"
    - John von Neumann in 1945

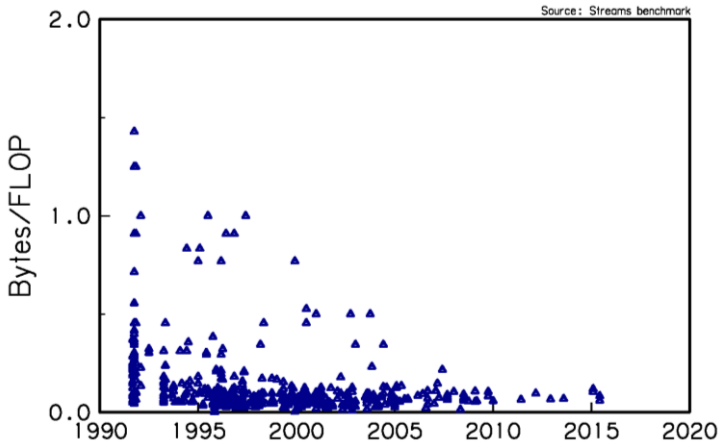
Balance and percent of peak

- Systems that have better “balance”, bytes/FLOP, deliver a greater percentage of peak
- Yes, it is as obvious as it seems



(Lack of) Balance over time

- There was a time in the past when there were systems with good balance
- Sadly there aren't any such systems any longer

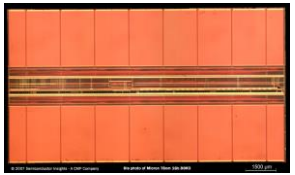


DRAM commonalities

- During DRAM's entire existence, 40+ years, there have been three immutable constants
  - It is volatile
  - It has been treated as a limited, expensive resource
  - It uses a lot of pins (~150 per channel)
- For the past ~22 years another commonality is that "DDR" and DRAM have been in a synergistic, inter-dependent embrace
  - This is good, but mostly bad

IBM 8 Mbit core
circa 1968

Intel 1103 DRAM
1k bit
circa 1970



Today's 8 Gbit DRAM die

DRAM

- DRAM is approaching its end of life or end of scaling
  - Does not mean it won't be made and sold
  - Capacity, efficiency, and cost/bit will plateau (or get worse)
- Something will replace DRAM
  - Almost certainly it will be nonvolatile memory (NVM)
  - The transition to NVM will be very long-tailed
  - DRAM in 2020+?
    - Certainly

Memory capacity today

- Memory capacity is dictated by
  - Number of DDR channels
  - Capacity of DRAM parts
  - Number of DIMMs
- Sockets are pin limited (and will remain so) hence the maximum number of DDR channels per socket is ~6 (4 is more common)
- Number of DIMMs per channel is limited to ≤4
  - Each additional DIMM causes the DDR channel to run at lower frequency
  - Forces a trade-off of capacity for performance, or vice versa

Desired/target NVM characteristics

- What we may expect from NVM (not the first generation) when compared to DRAM
  - Price per bit ~(1/10)
  - Bits per die >8x (for same size die!)
  - Energy per bit <(1/2)
  - Static power <(1/10)
  - Read latency <2x
  - Write latency <4x (but mostly hidden)

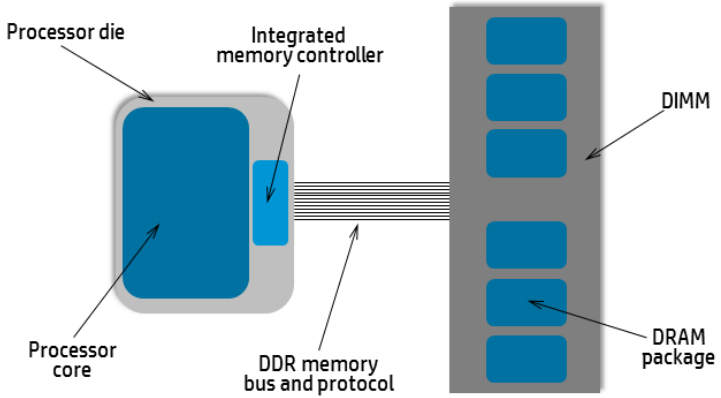
Memory-semantics (everywhere?)

- Just beginning to contemplate the extensive use of memory-semantics
  - The processor only emits ld/st requests
  - Fewer busses from the processor
  - Saves pins (i.e., power, cost, and silicon real estate)
- NICs and HDD/SDD devices will exist for a long time
  - Will, hopefully, exist at the end of a memory-semantic channel

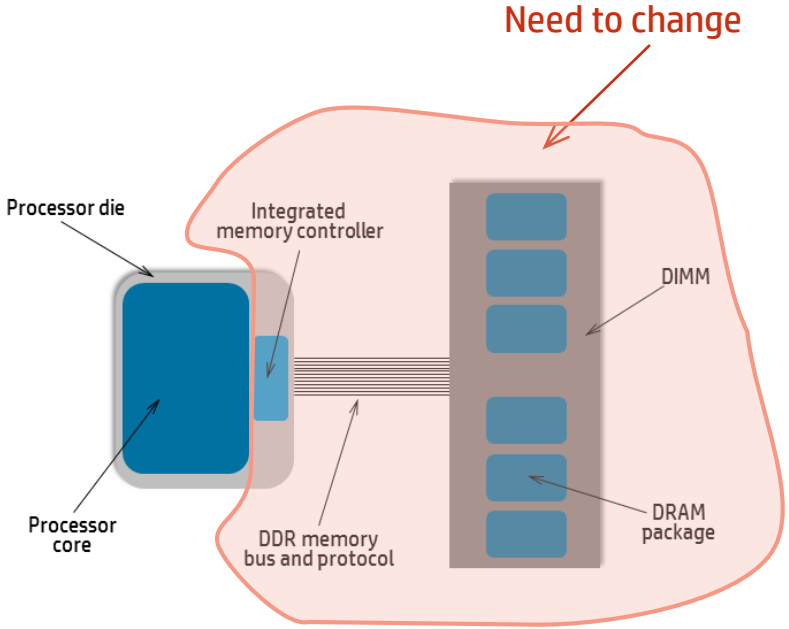
Physical address spaces

- Today's physical memory is measured in Gigabytes (2^{30})
  - End-users think, design, and code accordingly
- Tomorrow's physical memory will be Terabytes (2^{40})
  - Contemplate "storage" in the address space
  - Rethink algorithms and data structures
- (Tomorrow+ϵ)'s physical memory can be Petabytes (2^{50})
  - Eliminate all i/o operations, except archiving

What needs to change?



Today



Tomorrow

The remedy

Epilogue

- Shift R&D from FLOPS and OPS to communications and memory
  - Load/store semantics everywhere
  - New memory PHY and protocol
  - Transition to NVM
  - More physical address bits
  - Silicon nanophotonics, with DWDM