

# Macro Architecture Trends

Pavan Balaji

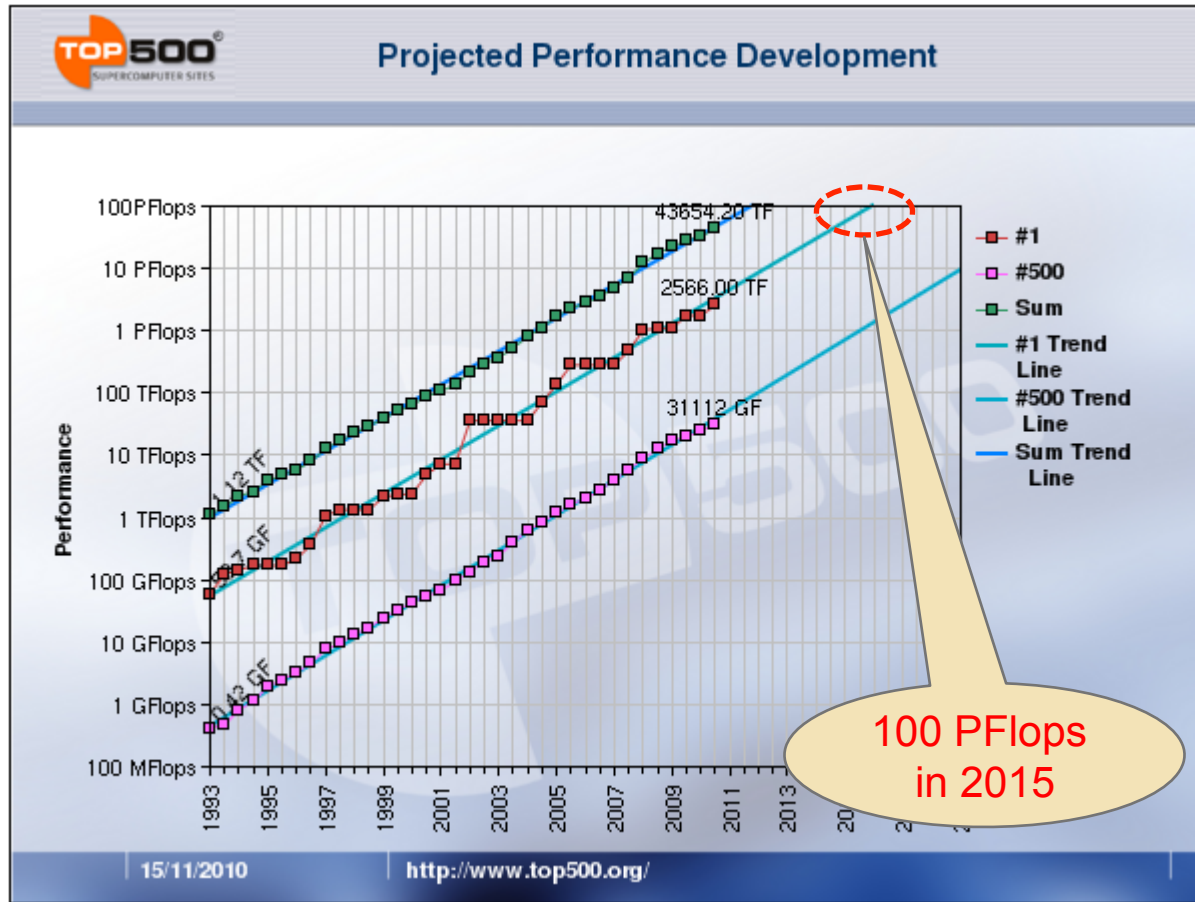
Computer Scientist

Programming Models and Runtime Systems Group

Mathematics and Computer Science Division

Argonne National Laboratory

# Trends in HPC



*Expected to have an Exaflop system in 2018-2020*



# Commodity Architectures in Scientific Computing

- High Performance Computing is not an economic driver
  - Vendors cannot make large profits selling only HPC specific machines
  - ... other than a few exceptions (such as Cray)
- For the past two decades, commodity hardware has been the primary driver for scientific computing
  - Hardware innovation was made profitable by desktop computers
    - Vendors could build faster processors, memory, etc., because they could make money out of it by selling it on desktops
  - There were, of course, HPC-specific pieces, but only a select few
    - Networks (Myrinet, Quadrics, InfiniBand)
    - ECC memory



# High-end Computing Systems Today

- Technological capabilities and scientific needs have triggered continuous growth in HPC capabilities
  - We grow by three orders-of-magnitude roughly every 11 years
- We have crossed the Petaflop barrier
  - Tianhe-1A, ORNL Jaguar, LANL Roadrunner
  - Argonne has a 10 PF BG/Q system
  - Lawrence Livermore has a 20 PF BG/Q system
- Exaflop systems will be out by 2018-2020
  - Expected to have of the order of a billion processing elements
    - Might be processors, cores, hardware threads
  - Nightmare to deal with management, reliability and other issues
  - Many hardware/software challenges to be addressed to get there



# HPC: Driving Factors at each Scale

- Pre Terascale: Computational Power
  - Faster processors
  - Commodity driven -- HPC-specific architectures too expensive to survive
- Terascale: Scalability
  - Communication/Networking models, debuggers and other tools
  - Mostly commodity – HPC-specific networks were innovated
- Petascale: Concurrency
  - Concurrency within the node: the multi-core era
  - Commodity market was a late arriver, but still the driver
- Exascale: Energy
  - We are close to the budget limits of centers that can deploy and run these systems (power cost has almost surpassed acquisition cost)
  - New innovations are required to meet this constraint
  - Will the commodity market pay for these innovations?



# U.S. DOE Potential System Architecture Targets

System attributes	2010	"2015"		"2018"	
System peak	2 Peta	200-300 Petaflop/sec		1 Exaflop/sec	
Power	6 MW	15 MW		20-30 MW	
System memory	0.3 PB	5 PB		32-64 PB	
Node performance	125 GF	0.5 TF	7 TF	1 TF	10 TF
Node memory BW	25 GB/s	0.1TB/sec	1 TB/sec	0.4TB/sec	4 TB/sec
Node concurrency	12	O(100)	O(1,000)	O(1,000)	O(10,000)
System size (nodes)	18,700	50,000	5,000	1,000,000	100,000
Total Node Interconnect BW	1.5 GB/s	20 GB/sec		200GB/sec	
MTTI	days	O(1day)		O(1 day)	

*Courtesy Kathy Yelick (Lawrence Berkeley National Laboratory)*



# Scale Changes Everything

	2010	2018	Factor Change
System peak	2 Pf/s	1 Ef/s	500
Power	6 MW	20 MW	3
System Memory	0.3 PB	10 PB	33
Node Performance	0.125 Gf/s	10 Tf/s	80
Node Memory BW	25 GB/s	400 GB/s	16
Node Concurrency	12 cpus	1,000 cpus	83
Interconnect BW	1.5 GB/s	50 GB/s	33
System Size (nodes)	20 K nodes	1 M nodes	50
Total Concurrency	225 K	1 B	4,444
Storage	15 PB	300 PB	20
Input/Output bandwidth	0.2 TB/s	20 TB/s	100

*Courtesy Dan Hitchcock (ASCR, U.S. DOE)*



# Power as a First Class Citizen

- Two primary constraints:
  - Upfront cost
  - Maintenance cost
- Power < 100MW (including cooling and storage)
  - 100-1000 times more power efficient than petascale systems
- Cost and Size
  - 100-1000 racks

## ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems

Peter Kogge, Editor & Study Lead  
Keren Bergman  
Shekhar Borkar  
Dan Campbell  
William Carlson  
William Dally  
Monty Denneau  
Paul Franzon  
William Harrod  
Kerry Hill  
Jon Hiller  
Sherman Karp  
Stephen Keckler  
Dean Klein  
Robert Lucas  
Mark Richards  
Al Scarpelli  
Steven Scott  
Allan Snavely  
Thomas Sterling  
R. Stanley Williams  
Katherine Yelick



September 28, 2008

This work was sponsored by DARPA IPTO in the ExaScale Computing Study with Dr. William Harrod as Program Manager; AFRL contract number **FA8650-07-C-7724**. This report is published in the interest of scientific and technical information exchange and its publication does not constitute the Government's approval or disapproval of its ideas or findings

### NOTICE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

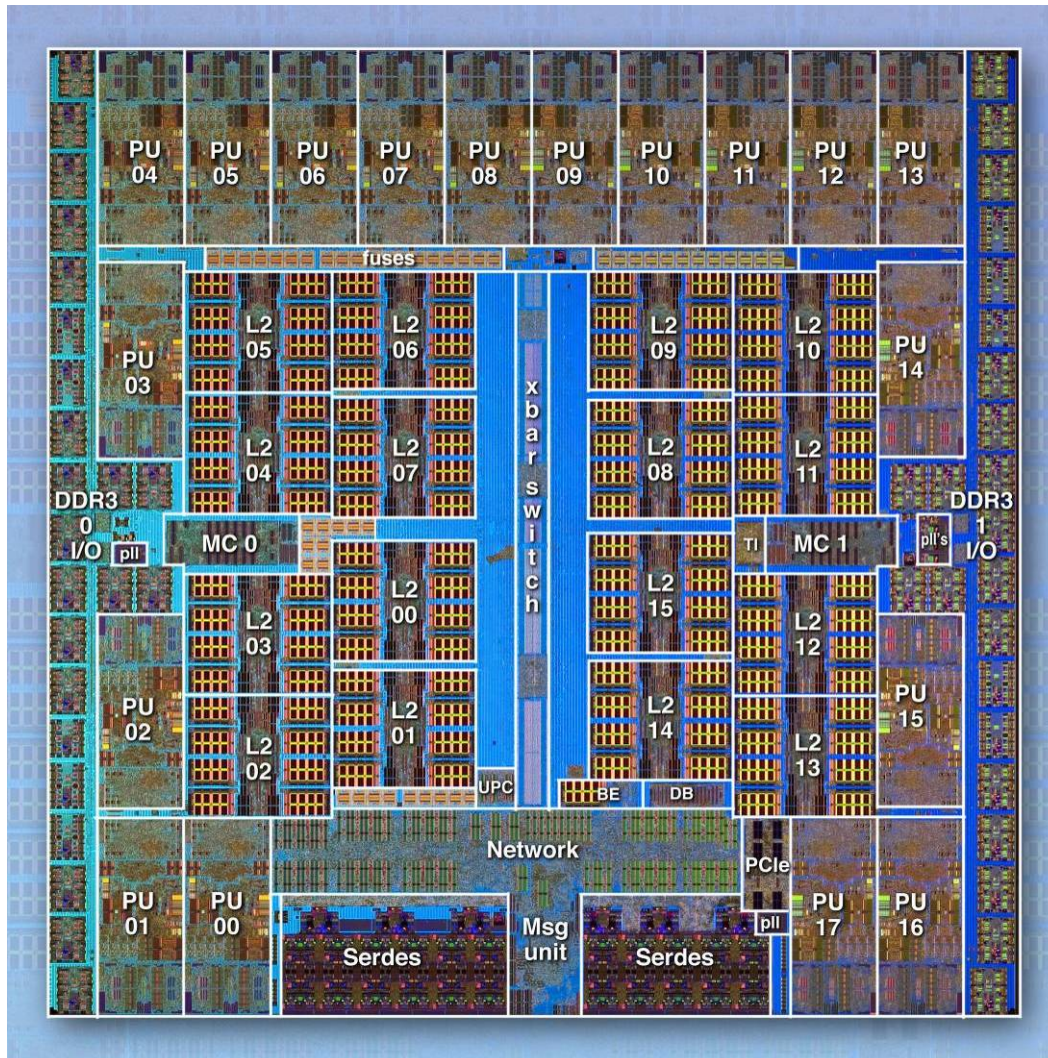
APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED.





# Processor Trends

# Blue Gene/Q Compute chip

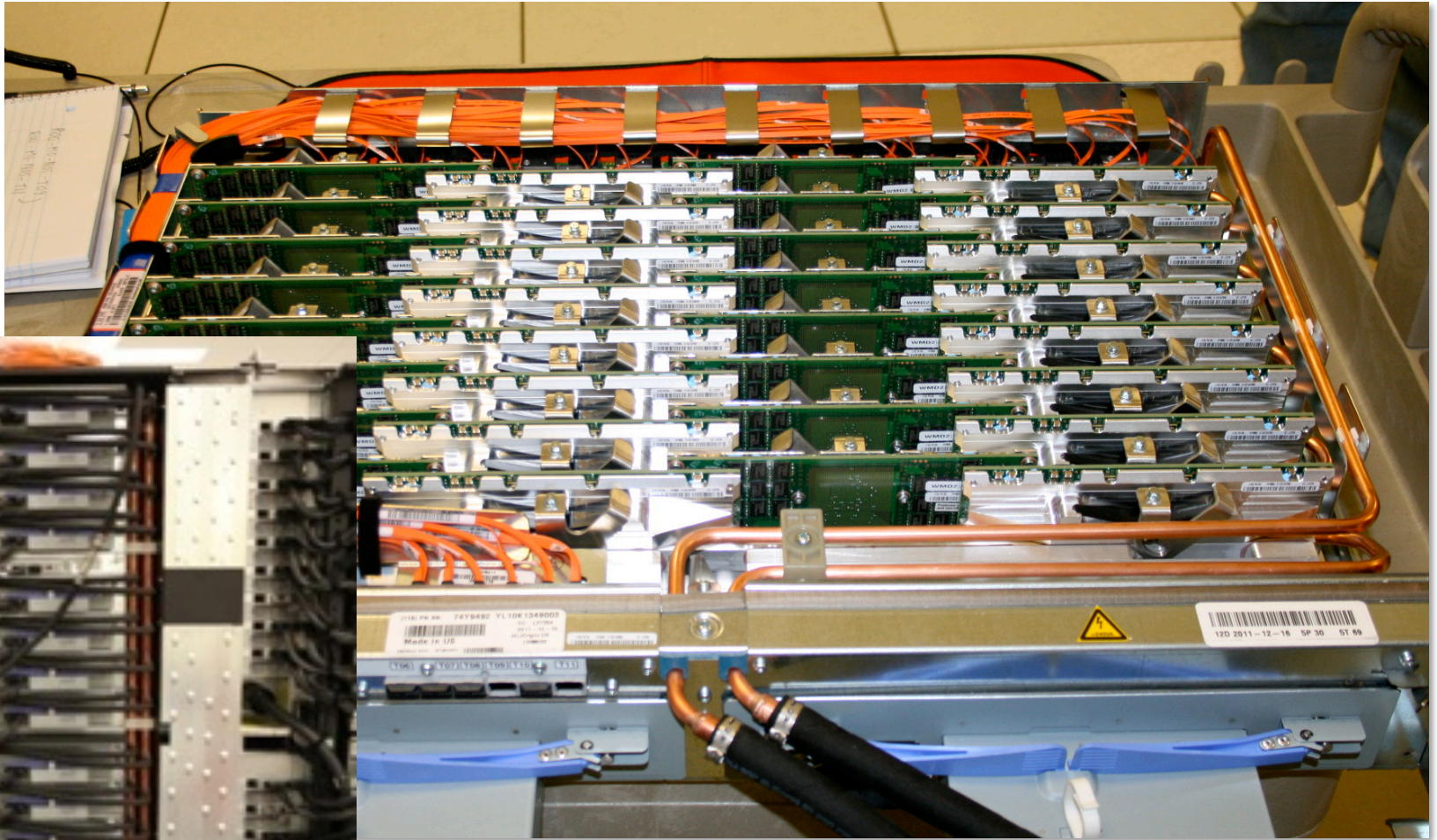


*Courtesy Pete Beckman (Argonne National Laboratory)*

- **360 mm<sup>2</sup> Cu-45 technology (SOI)**
  - ~ 1.47 B transistors
- **16 user + 1 service PPC processors**
  - plus 1 redundant processor
  - all processors are symmetric
  - each 4-way multi-threaded
  - 64 bits
  - 1.6 GHz
  - L1 I/D cache = 16kB/16kB
  - L1 prefetch engines
  - each processor has Quad FPU (4-wide double precision, SIMD)
  - peak performance 204.8 GFLOPS @ 55 W
- **Central shared L2 cache: 32 MB**
  - eDRAM
  - multiversioned cache – will support transactional memory, speculative execution.
  - supports atomic ops
- **Dual memory controller**
  - 16 GB external DDR3 memory
  - 1.33 Gb/s
  - 2 \* 16 byte-wide interface (+ECC)
- **Chip-to-chip networking**
  - Router logic integrated into BQC chip.
- **External IO**
  - PCIe Gen2 interface

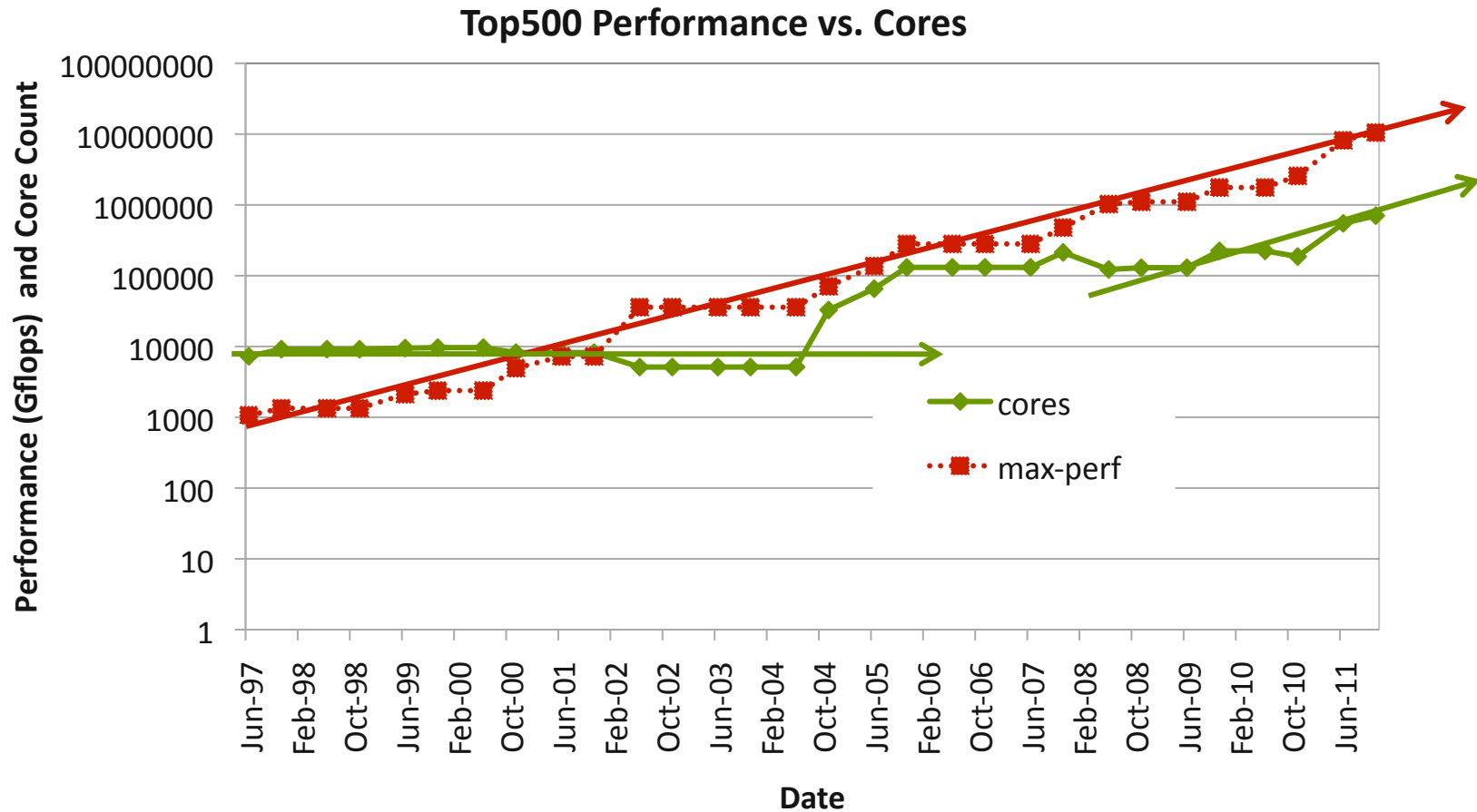






*Courtesy Pete Beckman (Argonne National Laboratory)*

# Concurrency is a Key Ingredient



*Clock speeds were the primary contributor to performance till about 2004. After that performance growth has closely matched the growth in concurrency. From here on, it is expected that concurrency will grow faster than the system performance.*



# Power will limit “full concurrency”

- Option 1: Heavily hierarchical processor architectures
  - The IBM EX4 and EX5 servers are examples of what is to come
    - Each “node” contains of multiple cache-coherent memory domains
    - Each memory domain contains multiple sockets (possibly NUMA)
    - Each socket contains multiple dies packaged together
    - Each die contains multiple cores
    - Each core contains multiple hardware threads (SMT)
  - Primary power saving comes from hardware sharing and packaging density (more density means lesser movement of data)
    - At each level in the hierarchy there is an increasing amount of hardware sharing (e.g., caches, memory controller units)
    - Locality will mean everything for performance!



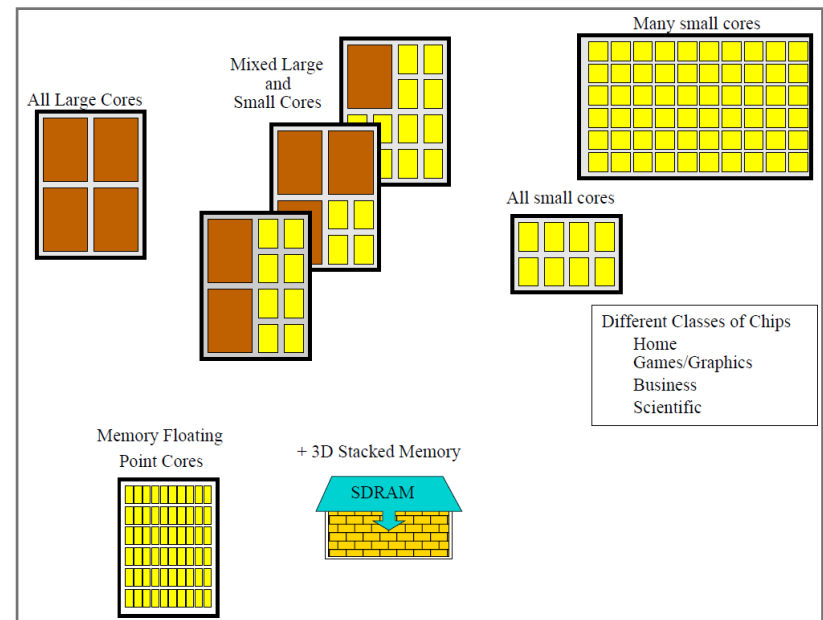
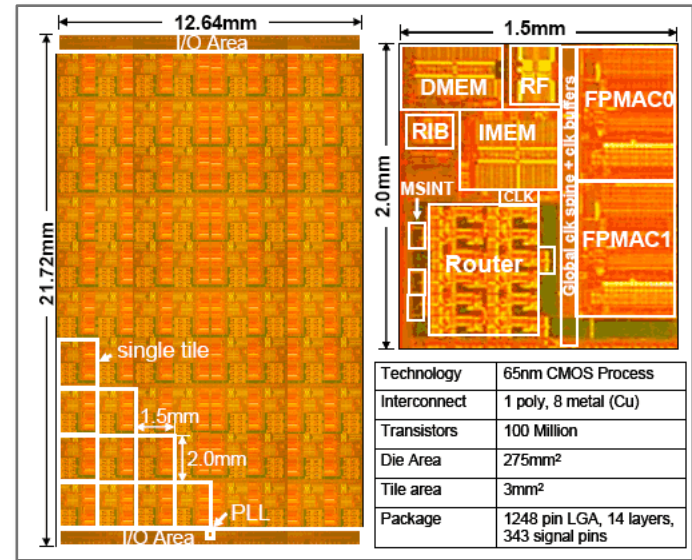
## Power will limit “full concurrency” (contd.)

- Option 2: Heterogeneous processor architectures
  - Graphics Processing Units (GPUs) are an example of what is to come
  - Primary power saving comes from reduced hardware units
    - Cache coherence might not be an option
      - No snooping hardware; everything might need to be explicitly managed
      - Explicit caches/scratchpad memory -- no unnecessary data movement (5% more data in cache block means 5% more power)
    - Shared double precision floating point unit for multiple threads
      - Floating point intensive applications will suffer
      - Increasing gap in single precision vs. double precision floating point OPs
    - Shared instruction decoder for multiple threads
      - Only suitable for extremely data parallel architectures
    - In order execution (no speculation hardware)



# Option 3 (the most likely possibility)

- Combination of Options 1 and 2
  - Lots of light-weight cores
    - In-order execution (no speculation hardware)
    - Small caches
    - No hardware cache coherency
  - Heterogeneous cores
    - General purpose processors + GPGPUs
    - More tightly integrated heterogeneous processors are coming
- Different cores can be switched on/off depending on application needs



*Courtesy William Gropp (UIUC)*





# Important concern: Increasing Divide in Processor Architecture Requirements

- None of the three options are really suitable for the commodity market (think “gamers” or “tablets”)
- What are the killer commodity market applications for these architectures?
  - Heavily threaded SMT architectures or deep hierarchical processors are really only meant for HPC (scientific computing and enterprise datacenters)
  - Accelerators/GPUs have good use cases in the commodity market, but HPC is driving a divide there too
    - Double precision floating point, ECC protection for GPU memory
- Integrated CPU+GPU architectures do have a market outside HPC, mainly in laptops/tablets (battery life)

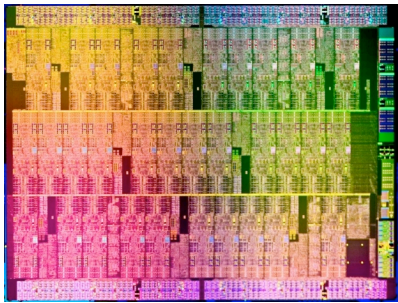


# Commodity Market Trends for Processors

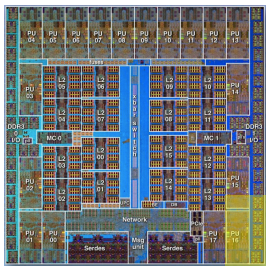
- General purpose processors will continue to be a major driving factor for the commodity mass market
  - “Good enough” computing – new trends are towards tablets and cell phones; don’t need more processing power, need power efficiency
- Increasing push towards processor specialization, but in ways HPC might not care about
  - Accelerator cores
    - Commodity market wants it for animations, movie rendering, games
    - Double precision floating points will not sell
    - ECC memory will not sell
  - Specialized hardware for other capabilities
    - Media decoders
    - Memory compression/decompression hardware



# Key Changes: Coherency, Power Management, Specialization



Intel: MIC

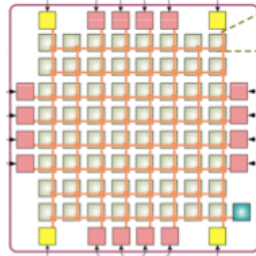


IBM: BG/Q

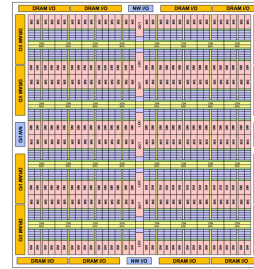
#18



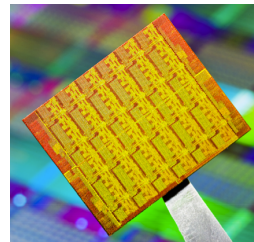
Power Constrained Memory Consistency



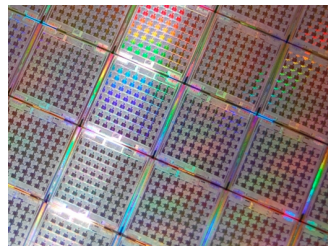
Godson T



Dally: Echelon



Intel: SCC



Tiler: GX

Extreme Specialization and Power Management

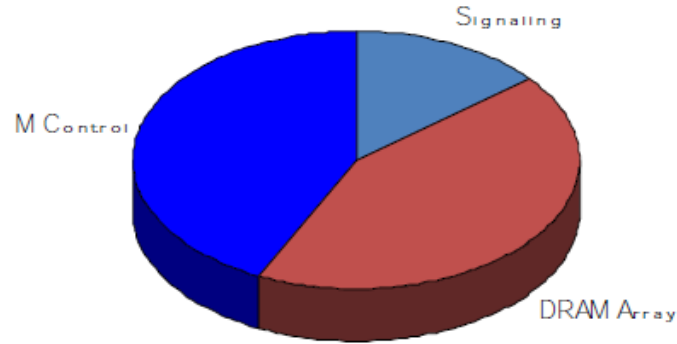


Chien: 10x10



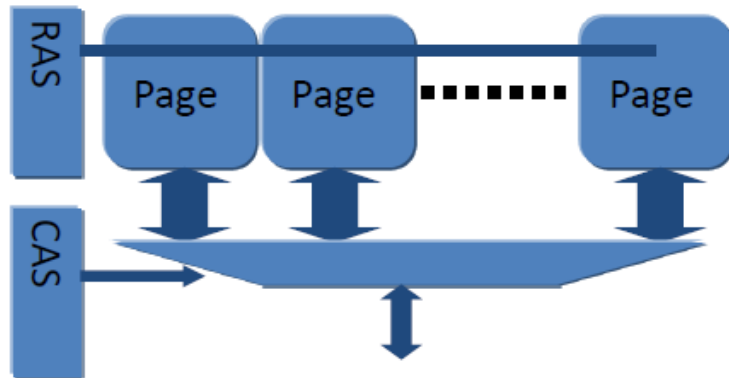
# Memory Trends

# Memory Architecture



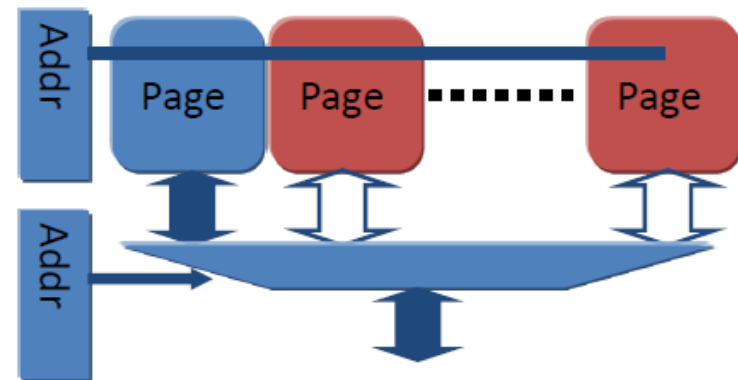
*Today's energy cost :  
~150 pJ/bit*

Today's DRAM



Activates many pages  
Lots of reads and writes (refresh)  
Small amount of read data is used  
**Requires small number of pins (DIPs)**

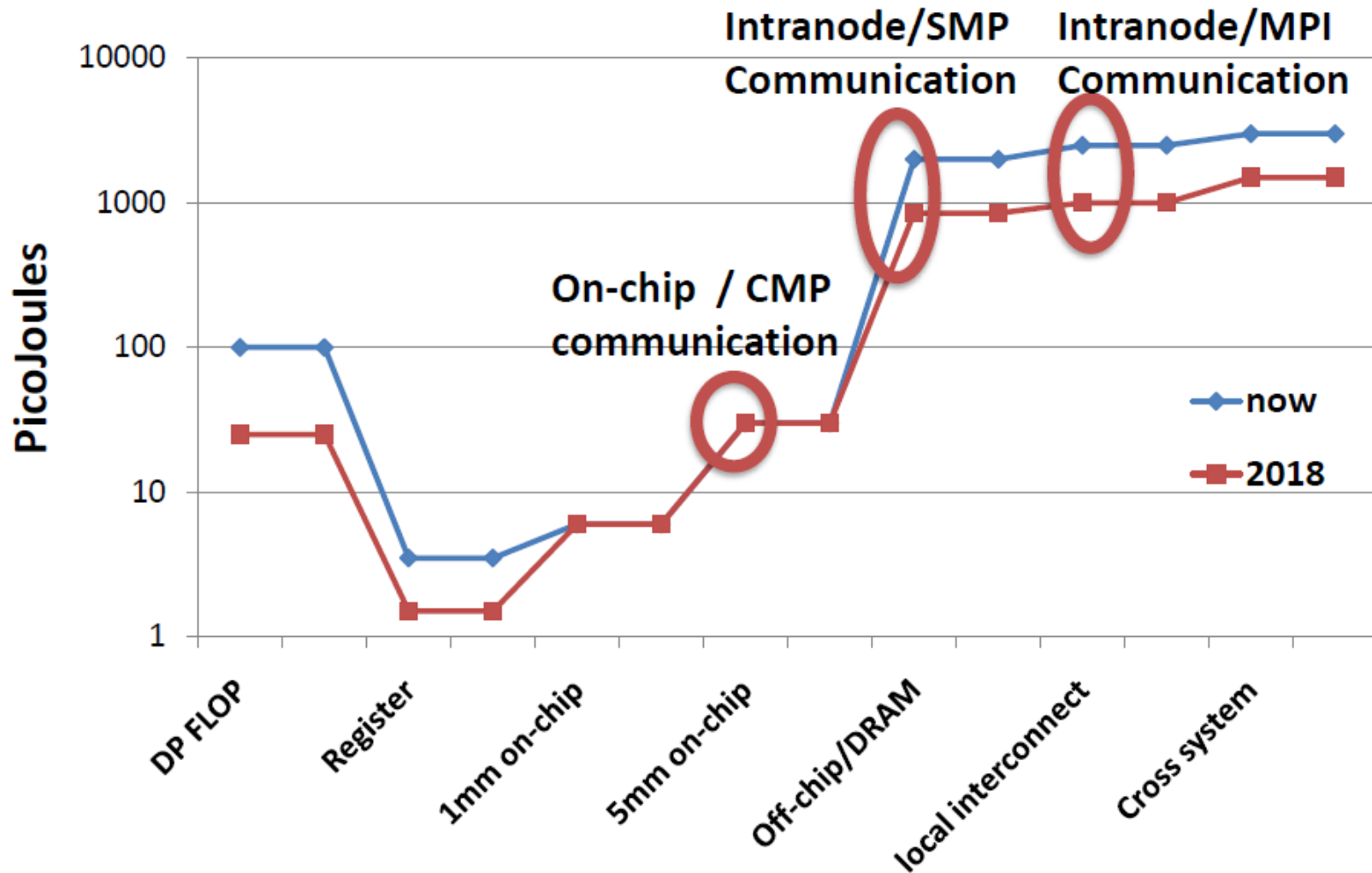
Tomorrow's DRAM



Activates few pages  
Read and write (refresh) what's needed  
All read data is used  
**Requires large number of IO's (3D)**



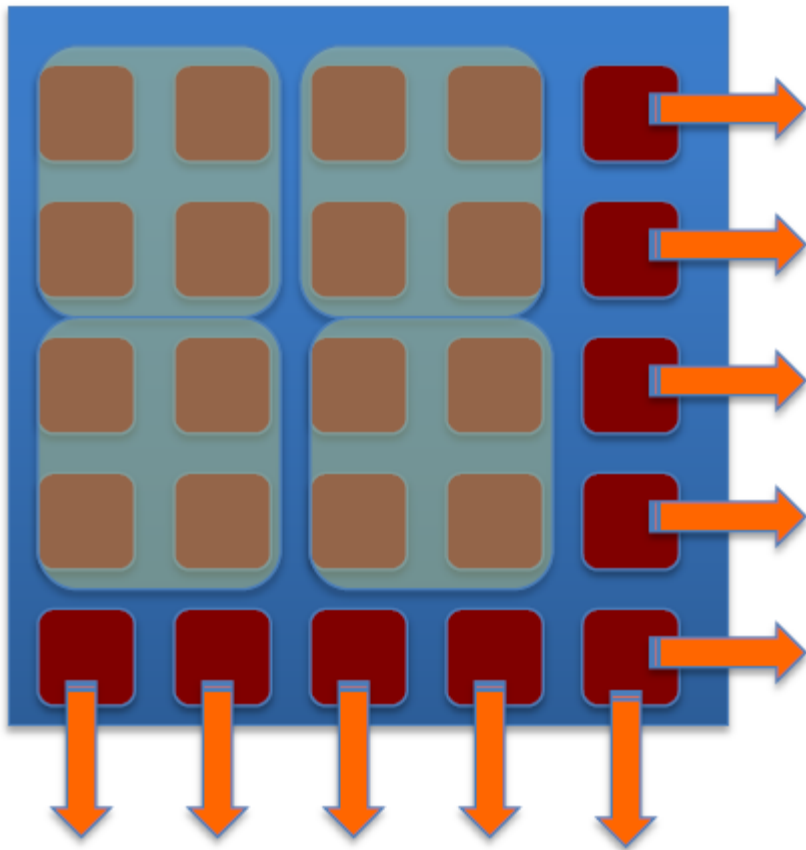
# Data Movement is Expensive



*Courtesy John Shalf (Lawrence Berkeley National Laboratory)*

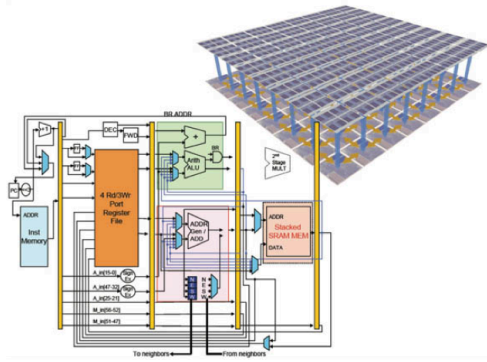


# Processor Density as a path to Data Locality

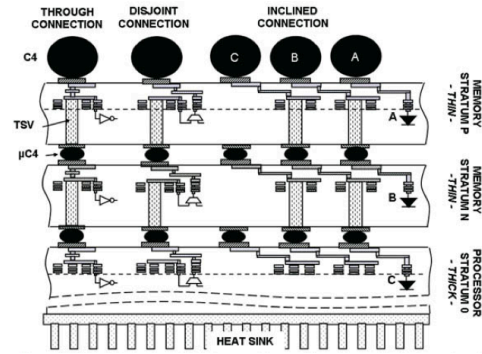


- Cost of moving data
  - 1mm costs  $\sim 6\text{pj}$  (today & 2018)
  - 20mm costs  $\sim 120\text{pj}$  (today & 2018)
  - FLOP costs  $\sim 100\text{pj}$  today
  - FLOP costs  $\sim 25\text{pj}$  in 2018
- Heavy processor density within the chip might be the only way to reduce inter-process data movement energy costs

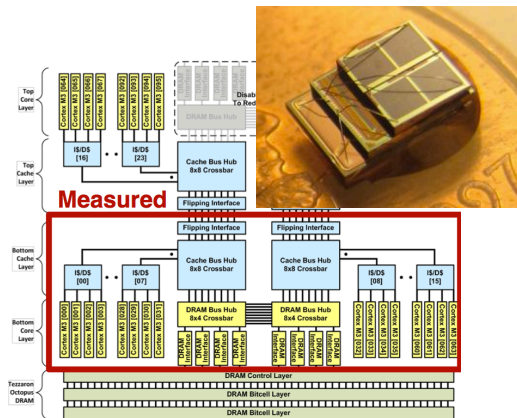
# 3D Chip Stacking: Fast, Close, (relatively) Small



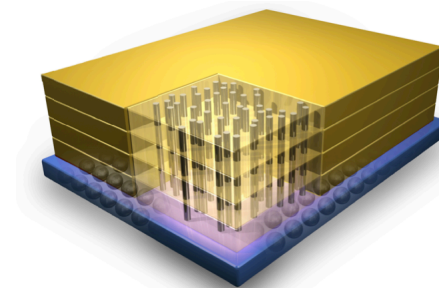
Georgia Tech



IBM



Univ of Michigan



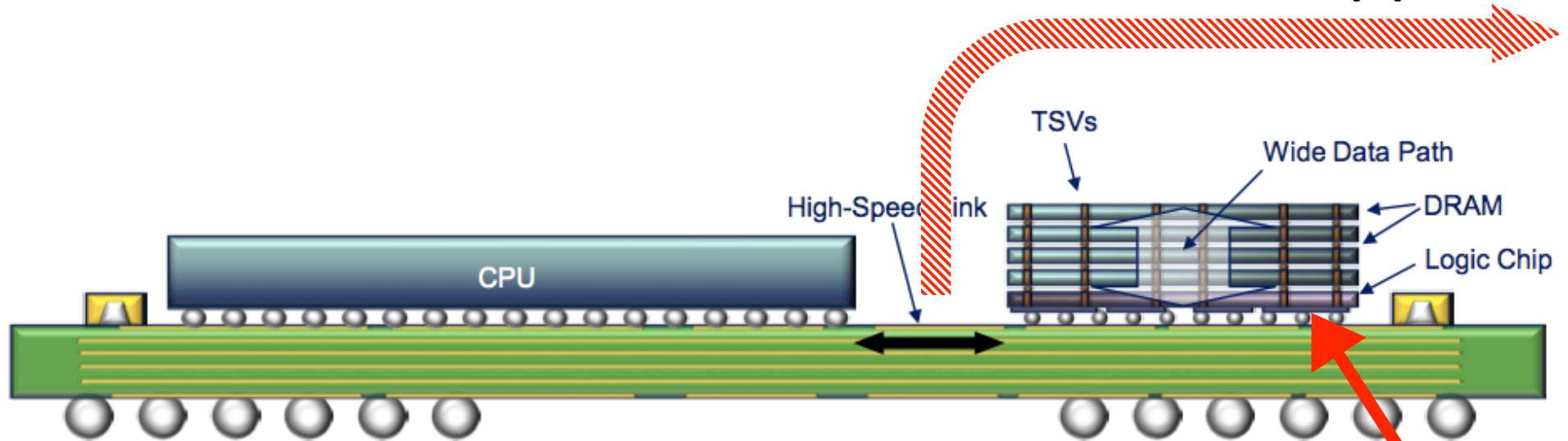
Micron HMC





# Micron Hybrid Memory Cube

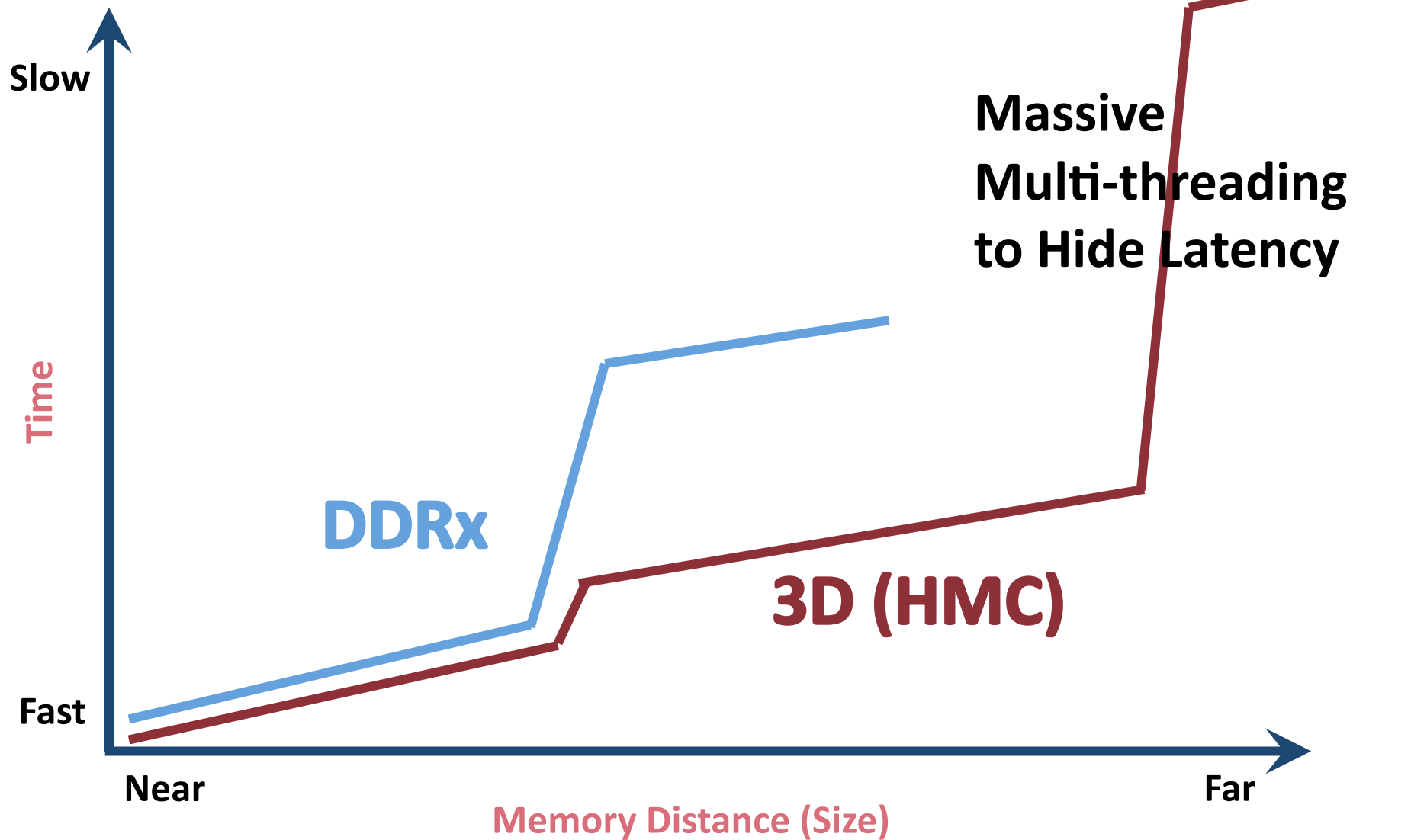
Future on-module  
Interconnect pipe?



“Early benchmarks show a memory cube blasting data 12 times faster than DDR3-1333 SDRAM while using only about 10 percent of the power.”



# HMC Projections



## 3D Stacked Memory

- Allows memory to be close to the processing units
  - Can be a separate die (connected directly via die pins, rather than through an I/O hub)
    - No memory bus overhead
  - Can be integrated on the same chip (on-chip memory)
    - No die pin overheads for data movement
  - Very NUMA and NUCA, but data access costs are much smaller
- Complex and expensive chip architectures
- Something has to give:
  - Increasing density means lesser data movement, but more power as well (die area is not as much of a problem as it used to be)
  - Vendors looking at “dark silicon” techniques, where the chip can contain many units, but only some of them can be turned on at a given time



# Implicit Caches vs. Scratchpad Memory

- Implicit caches have long served processor architectures as a way to transparently make memory look faster
  - When a data element is accessed, it is fetched to cache (together with a bunch of additional data, which the caching logic believes will be accessed next)
  - When there is no more extra space in the cache, something is kicked out back to memory
- Very convenient to use, but their efficiency depends on how data is accessed (because of the extra data fetched on a miss)
  - Many techniques to improve cache efficiency over the years, but not quite sufficient for all applications
  - Moving useless data means wasted energy – cannot afford at exascale
- Architectures are moving towards “explicit caches” or scratchpad memory



# Memory Reliability

- HPC already has a divide from the mass market with respect to memory reliability requirements
  - Mass market does not really care about memory reliability (“unreliable” memory DIMMS are still quite reliable)
    - A bit flip in memory would mean that your Windows server crashes once every year or your game has a small hiccup
  - For HPC, a bit flip means that the results are unreliable and redundant compute power has to be used to ensure the results are correct
    - HPC market has pushed the need for hardware error coding and correction (ECC memory)
- At Exascale, even ECC might not be sufficient
  - ECC can protect against single bit flips, not double bit flips
  - Oak Ridge is already seeing double bit flips once a week on their Jaguar system – at Exascale, we will be seeing more



# Memory Reliability Options

- Option 1: Techniques available for higher memory reliability
  - 2D error coding techniques available for protecting against double bit flips – not a technological limitation
  - But will the mass market care about this? They don't even care about ECC yet
  - If the mass market won't pay for this, the cost of doing this for HPC will be staggering!
- Option 2: Vendors are also considering having two types of memory: “more reliable (using higher redundancy)” and “less reliable (using ECC)”
  - HPC applications might have to deal with what memory they put what data on
- Neither option is really a seller for the mass market!



# Memory Consistency Semantics

- Weak memory consistency across parallel processors is a well understood problem
  - If one process writes data to a global memory location, another process cannot read it without appropriate explicit synchronization
- ... but for a single processor, we tend to assume strong memory consistency
  - `A = 1 ; X = A ; assert(X == 1) ;`
  - Not a good assumption anymore!
    - Just because one core looks/feels serial does not mean that it is serial
    - Each core has a lot of redundant hardware for internal parallelism – vector instructions, out-of-order instructions using redundant hardware, multiple paths to memory



# Memory Consistency Semantics Discrepancy

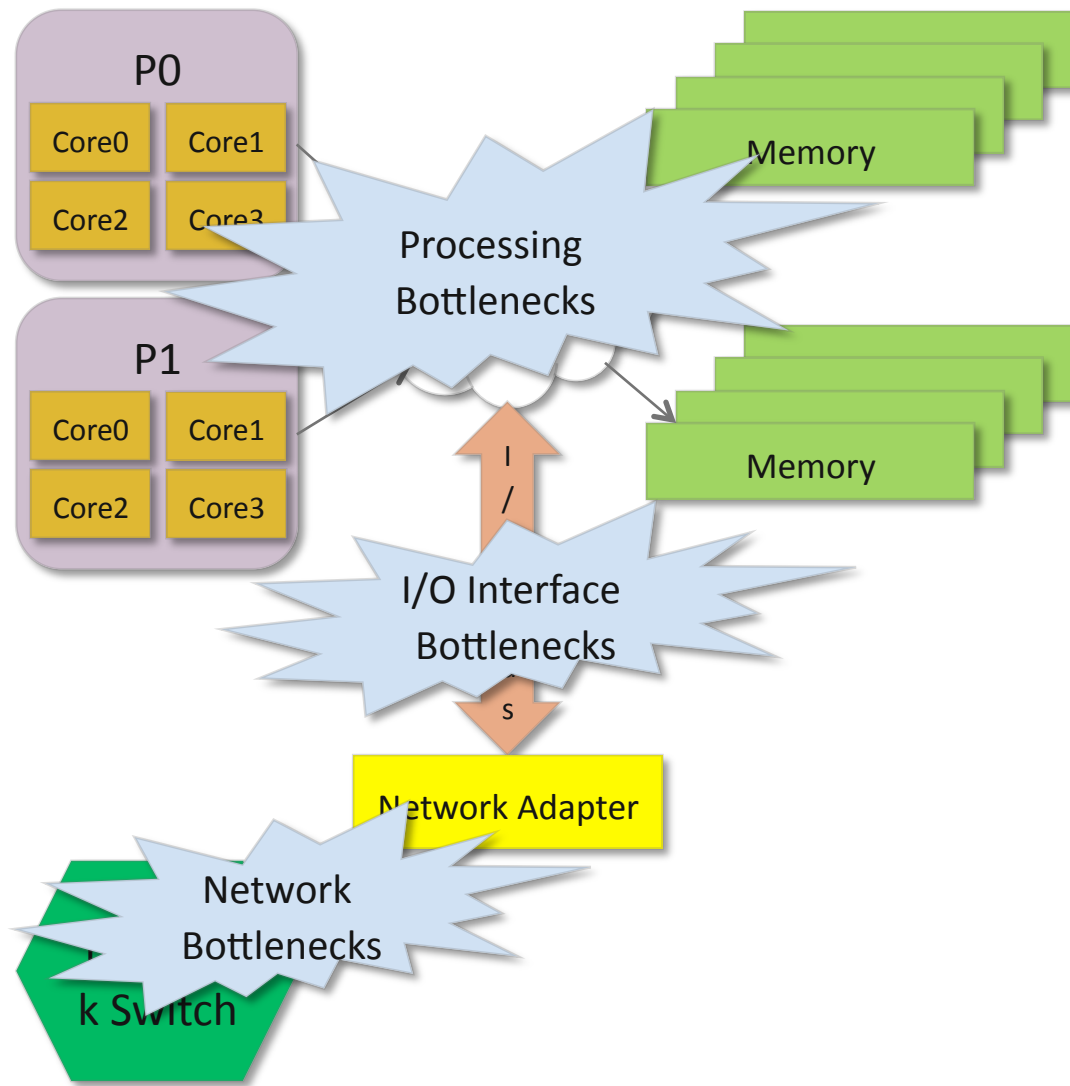
- However, commodity processors go out of their way to provide strong memory consistency
  - OK for the mass market, as the “lost” parallelism enforced by these semantics is not enough of an overhead to justify rewriting billions/trillions of dollars of software
- HPC processor variants are looking at providing weaker memory consistency semantics to fully utilize the internal parallelism in access to memory





# Network Trends

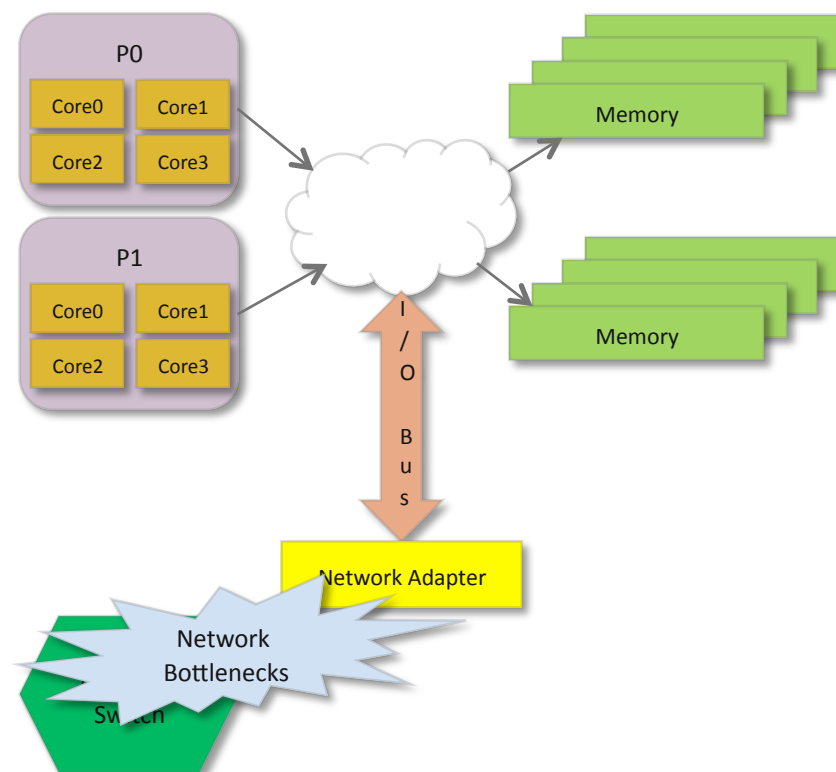
# Network Architecture Components



- Hardware components
  - Processing cores and memory subsystem
  - I/O bus or links
  - Network adapters/ switches
- Software components
  - Communication stack
- *Balanced approach required to maximize user-perceived network performance*

# Bottlenecks on Traditional Network Raw Speeds

- Network speeds saturated at around 1Gbps
  - Features provided were limited
  - Commodity networks were not considered scalable enough for very large-scale systems

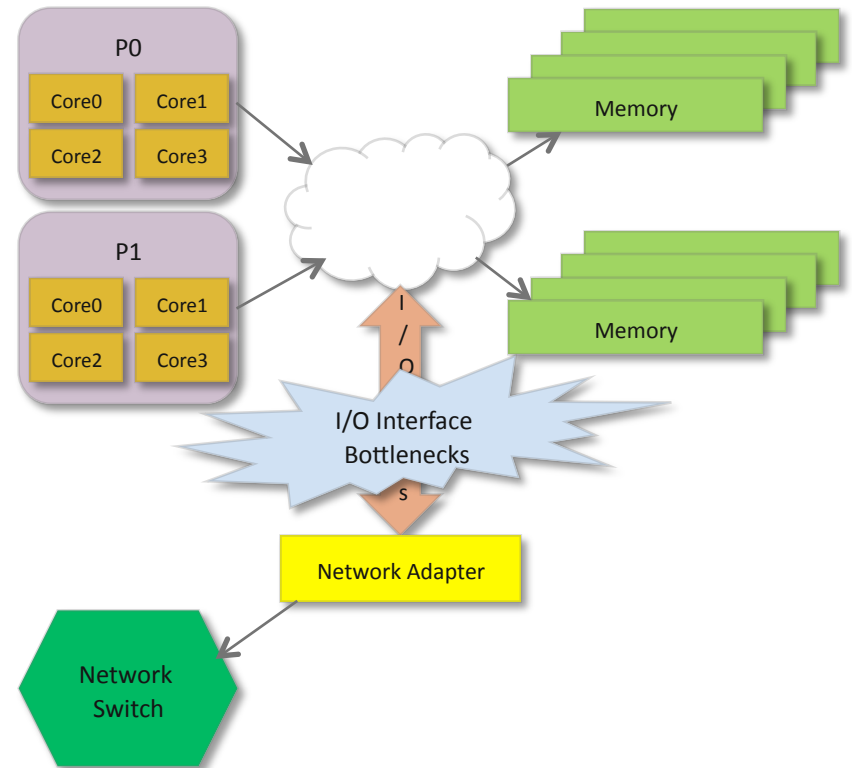


Ethernet (1979 - )	10 Mbit/sec
Fast Ethernet (1993 - )	100 Mbit/sec
Gigabit Ethernet (1995 - )	1000 Mbit /sec
ATM (1995 - )	155/622/1024 Mbit/sec
Myrinet (1993 - )	1 Gbit/sec
Fibre Channel (1994 - )	1 Gbit/sec



# Bottlenecks in Traditional I/O Interfaces and Networks

- Traditionally relied on bus-based technologies (last mile bottleneck)
  - E.g., PCI, PCI-X
  - One bit per wire
  - Performance increase through:
    - Increasing clock speed
    - Increasing bus width
  - Not scalable:
    - Cross talk between bits
    - Skew between wires
    - Signal integrity makes it difficult to increase bus width significantly, especially for high clock speeds

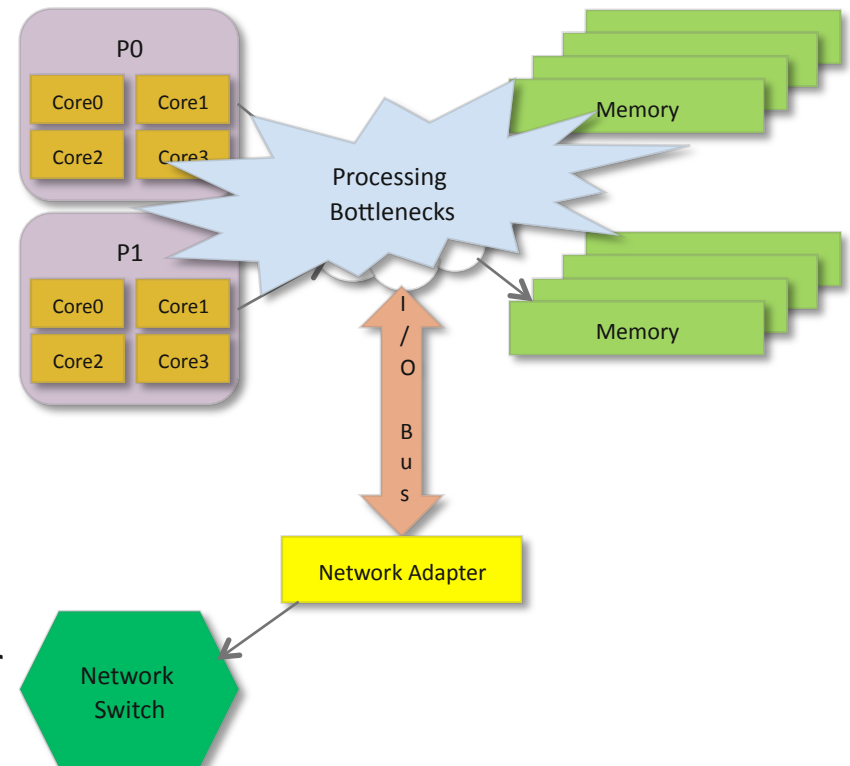


PCI	1990	33MHz/32bit: 1.05Gbps (shared bidirectional)
PCI-X	1998 (v1.0)	133MHz/64bit: 8.5Gbps (shared bidirectional)
	2003 (v2.0)	266-533MHz/64bit: 17Gbps (shared bidirectional)



# Processing Bottlenecks in Traditional Protocols

- Ex: TCP/IP, UDP/IP
- Generic architecture for all networks
- Host processor handles almost all aspects of communication
  - Data buffering (copies on sender and receiver)
  - Data integrity (checksum)
  - Routing aspects (IP routing)
- Signaling between different layers
  - Hardware interrupt on packet arrival or transmission
  - Software signals between different layers to handle protocol processing in different priority levels



# Network Technology Trends (1/3)

- Network Raw Speed
  - Recent network technologies provide high bandwidth links (e.g., InfiniBand FDR gives 14 Gbps per lane/56 Gbps per network link)
  - With multiple network links or network paths in Torus type of topologies, this bandwidth can be further increased
  - Network bandwidth “mostly” is no longer considered a major technological limitation (in some cases, it still is, because of network congestion)
  - Network latency is still an issue, but that’s a harder problem to solve



## Network Technology Trends (2/3)

- I/O Interface Speeds
  - Traditional I/O bus models (PCI, PCI-X) are long-gone at this point
  - Newer I/O interconnect models such as PCIe 1/2/3 and HTX available today
    - Theoretically infinite bandwidth because of multi-lane technologies (independent pairs of wires for each lane improves signaling efficiency and hence clock speeds)
    - Still a band-aid solution, but the most prominent band-aid solution today
  - Some vendors are looking at completely removing the I/O interconnect from the network critical path
    - Processor integrated networks, can have direct access to the memory controller and hence can move data directly from memory



# Network Technology Trends (3/3)

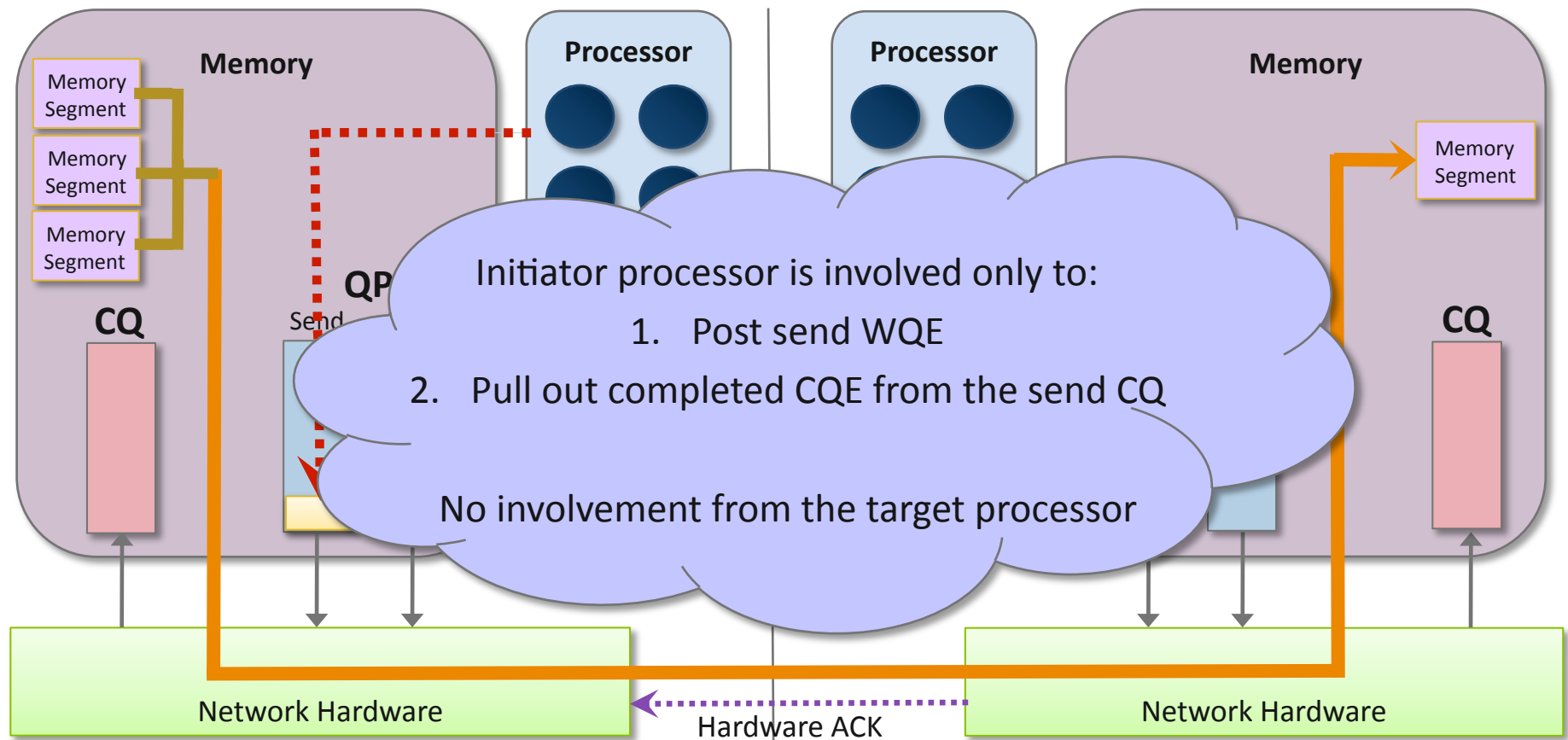
## ■ Network Protocol Stacks

- Modern networks are spending more and more network real-estate on offloading various communication features on hardware
- Network and transport layers are hardware offloaded for most modern networks
  - Reliability (retransmissions, CRC checks), packetization
- Some networks, such as the Blue Gene network, Cray network and InfiniBand, are also offloading some MPI and PGAS features on to hardware
  - E.g., PUT/GET communication has hardware support
  - Cray Seastar also had support for hardware matching for MPI send/recv
- Increasing network specialization is the focus today
  - The next generation of networks plan to have further support for noncontiguous data movement, and multiple contexts for multithreaded architectures





# Hardware supported PUT/GET Communication



Send WQE contains information about the send buffer (multiple segments) and the receive buffer (single segment)



## Summary

- With every 10X increase in performance, something breaks!
- In the past 20 years, we have enjoyed a million-fold increase in performance
  - Patch-work at every step is not going to cut it at this pace
  - We need to look forward for what's next in technology and think about how to utilize it
- We are looking at another 30X increase in performance over the next decade
- These are interesting times for all components in the overall system architecture: processor, memory, interconnect
  - And interesting times for computational science on these systems!



# Thank You!

Email: [balaji@mcs.anl.gov](mailto:balaji@mcs.anl.gov)

Web: <http://www.mcs.anl.gov/~balaji>

Argonne Programming Models Group

MPICH: <http://www.mcs.anl.gov/mpich2>

Radix Systems Software: <http://www.mcs.anl.gov/research/radix>