

# X-caliber HPC User Forum

**Arun Rodrigues**  
**Scalable Computer Architectures Department**  
**Sandia National Laboratories**



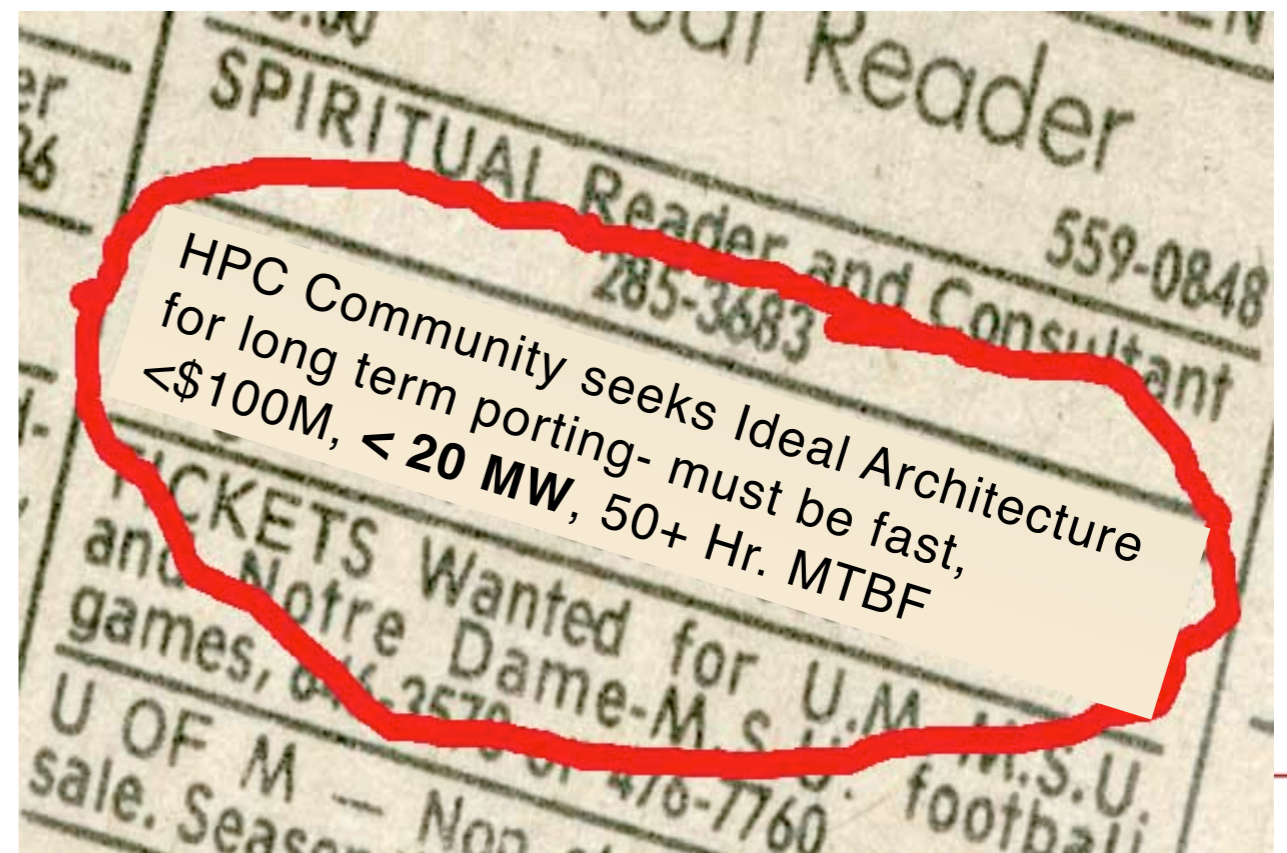
Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,  
for the United States Department of Energy's National Nuclear Security Administration  
under contract DE-AC04-94AL85000.

# Challenges for Exascale

- Can we get to Exascale?
- Technical Challenges
  - Power
  - Performance
  - Cost
  - Reliability
  - Programmability
  - Cooling
- Cultural Challenges
  - Programmability
  - Co-design
  - “The New COTS”: leveraging mobile & embedded IP

Scientific challenges such as understanding the causes and potential impacts of climate change, improving the efficiency of combustion, and unraveling the mysteries of dark energy and dark matter, as well as a variety of national security challenges, require computational capabilities at extreme scale...

DOE/SC DE-FOA-0000255



# Power is the Problem

## • 2018 Exascale Machine

- 1 Exaop/sec
- 100s petabyte/sec memory bandwidth
- 100s petabyte/sec interconnect bandwidth
- No major architecture changes

	Energy	Conventional
Processor	62.5 pJ/op	62.5 MW
Memory	31.25 pJ/bit	125 MW
Interconnect	6 pJ/bit	24 MW
<b>Total</b>		<b>211.5 MW</b>

## • Consider power

- 1 pJ \* 1 Exa = 1 MW
- 1 MW/year = \$1 M
- \$200-400M / year power bill

	2018 Estimate
Processing	224 MW
Memory	125 MW
Interconnect	24 MW
<b>Total</b>	<b>373 MW</b>



Component	Base (MW)
Memory BW	30
Memory Cap.	5.6
Network	17.2
Processor	22
<b>Total</b>	<b>74.8</b>

From Jensen "Embedded systems and exascale computing." CiSE 2010

# Worldwide Impact

**"Total power used by servers [in 2005] represented ... an amount comparable to that for color televisions. "**

**-ESTIMATING TOTAL POWER CONSUMPTION BY SERVERS IN THE U.S. AND THE WORLD, Jonathan G. Koomey**

<b>3741e9 KW-Hrs</b>	<b>Total US power consumption</b>
<b>* 3-4%</b>	<b>used by computers (&gt;2% servers, &gt;1% household computer use)</b>
<b>= 112 - 150e9 KW-Hrs</b>	<b>US Computer power consumption</b>
<b>* \$0.1 \$/KW-Hr</b>	<b>Retail cost, US Average 2009</b>
<b>= \$11 - \$15</b>	<b>Billion US\$ in compute power</b>
<b>* 3-5</b>	<b>in 2005 US was roughly 1/3 of servers, by power. This has probably decreased</b>
<b>= \$33 - \$75</b>	<b>Billion US\$ in worldwide computer power</b>
<b>=  - </b>	<b>Yearly GDP of Qatar to Burma</b>

# X-Caliber UHPC Project

## X-caliber Program Office

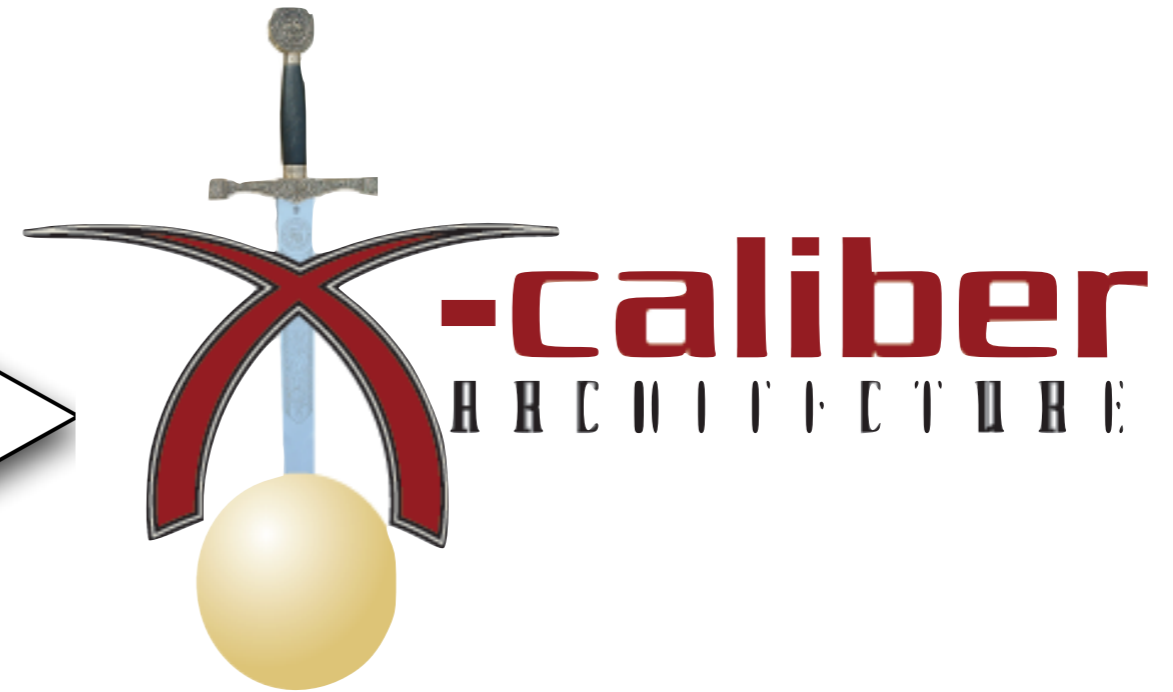
Principal Investigator: Richard Murphy  
Program Manager: Jim Ang  
Chief Scientist: Peter Kogge  
Senior Advisor: Jim Tomkins

**Programming  
Methods**  
Bill Gropp  
Mike Heroux

**Software  
Architecture**  
Marc Snir  
Ron Brightwell

**Hardware  
Architecture**  
Bob Lucas  
Scott Hemmert  
Jeff Draper

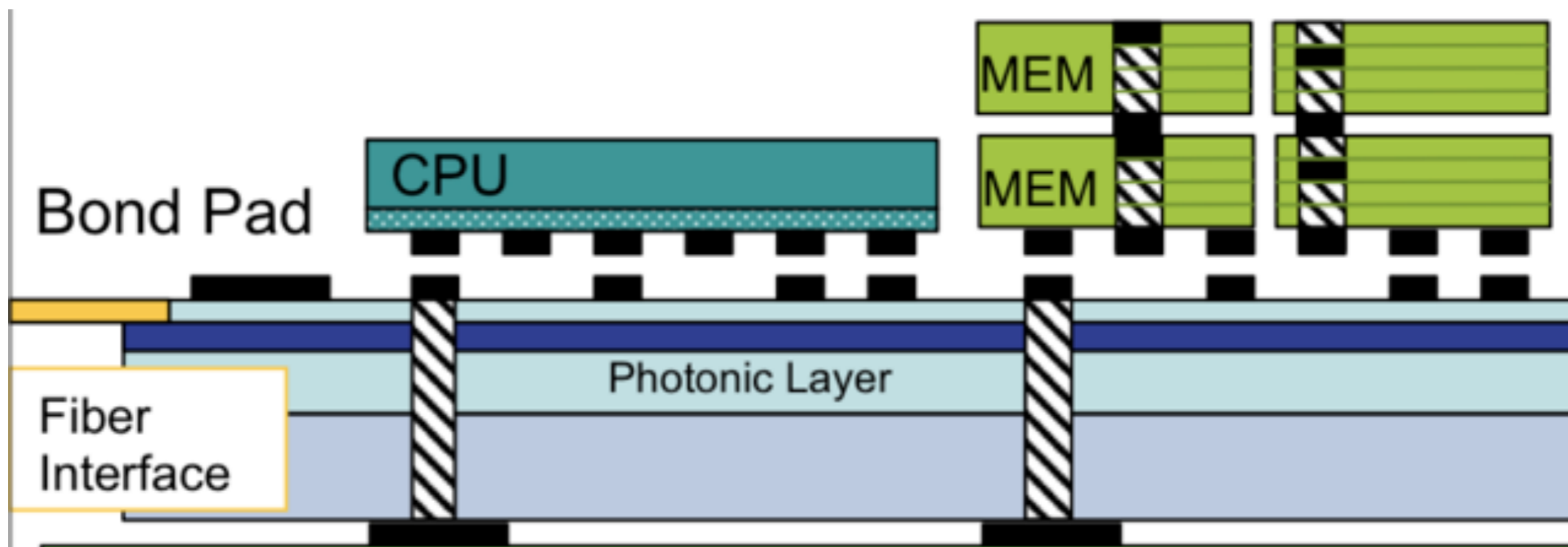
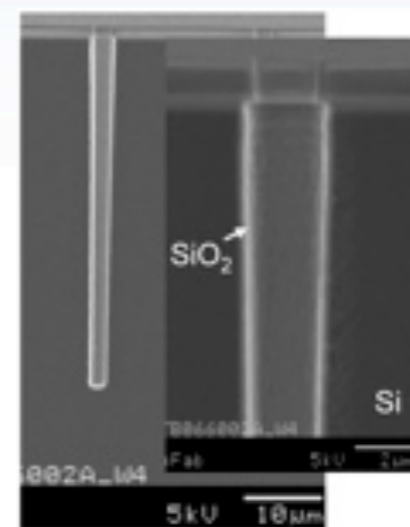
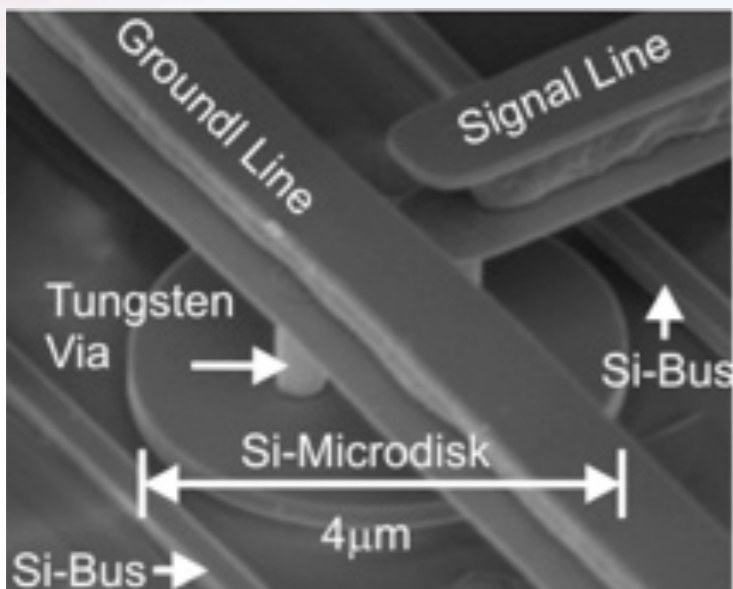
- **Focus Areas**
  - **Technology**
  - **Architecture**
  - **Execution Models**





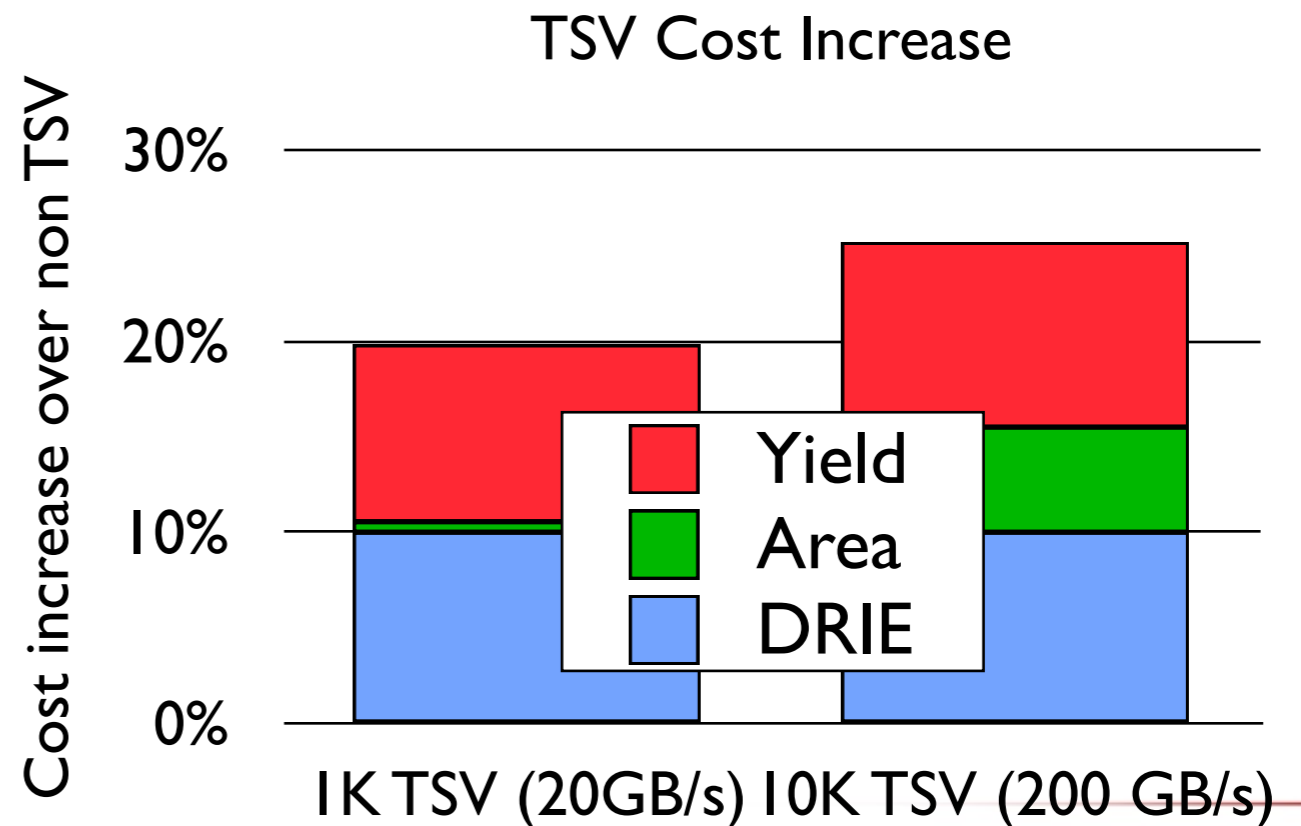
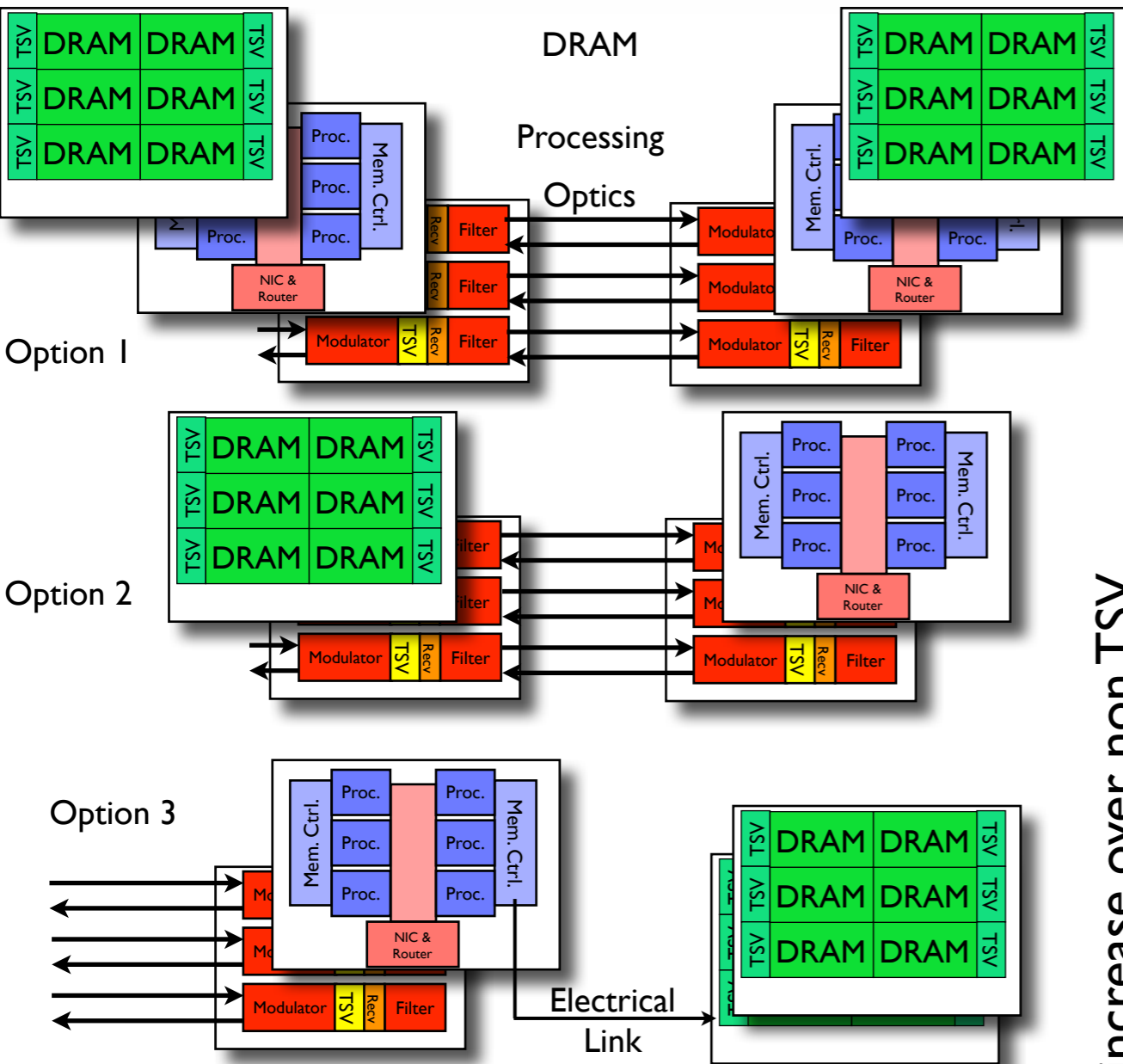
# Technology

# Our Enabling Technologies: Advanced Packaging, 3D Memory Integration, Si Photonic Communication



# Stacking!

- Enables integration with photonics
- Relatively low cost
- Huge amounts of bandwidth
- Many interesting research questions

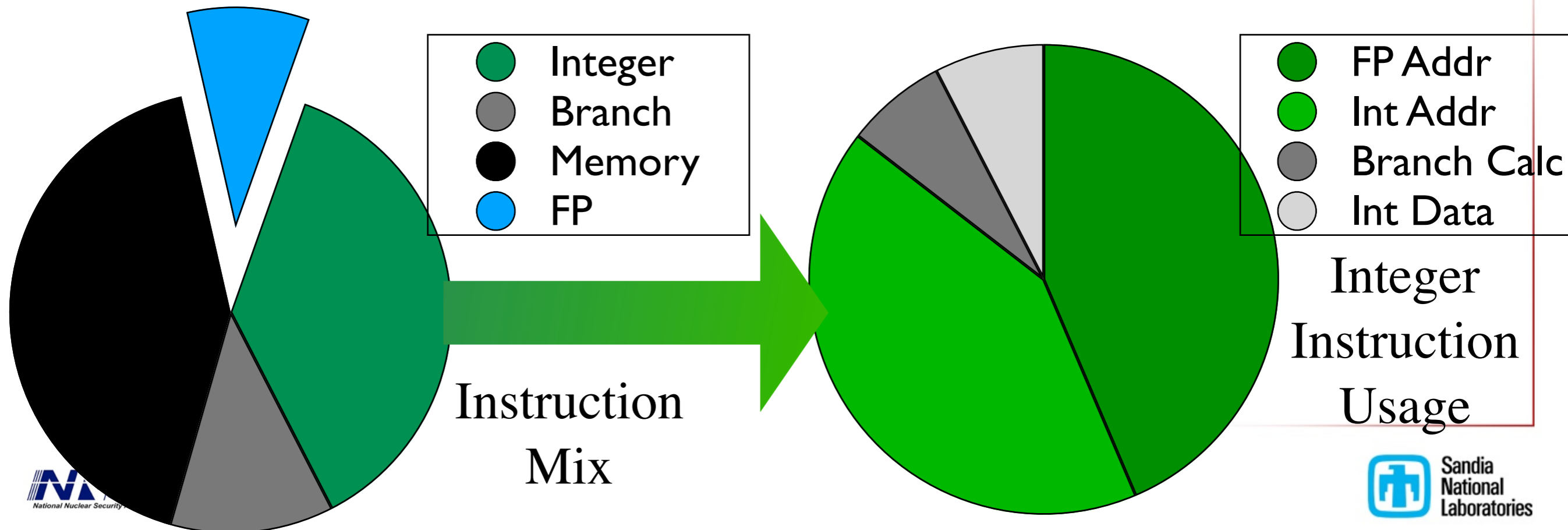
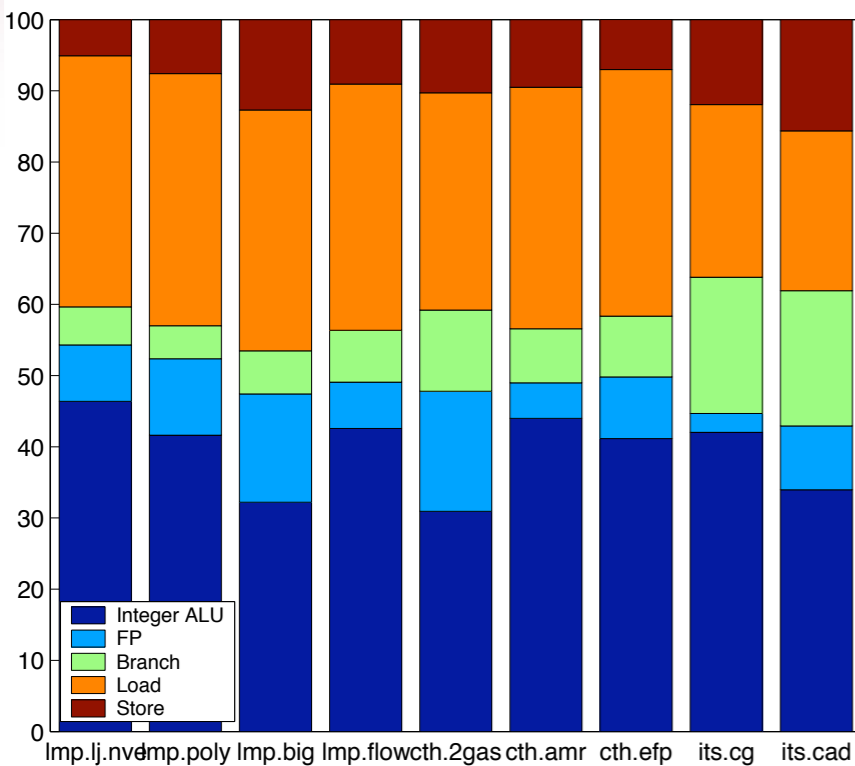




# Architecture

# Memory Operations Dominate On-Node Performance

- FP ops (“Real work”) < 10% of Sandia codes
- Several Integer calculations, loads for each FP load
- Memory and Integer Ops dominate
  - ...and most integer ops are computing memory addresses
- Theme: Even FP applications mainly do memory operations, or integer operations supporting memory operations



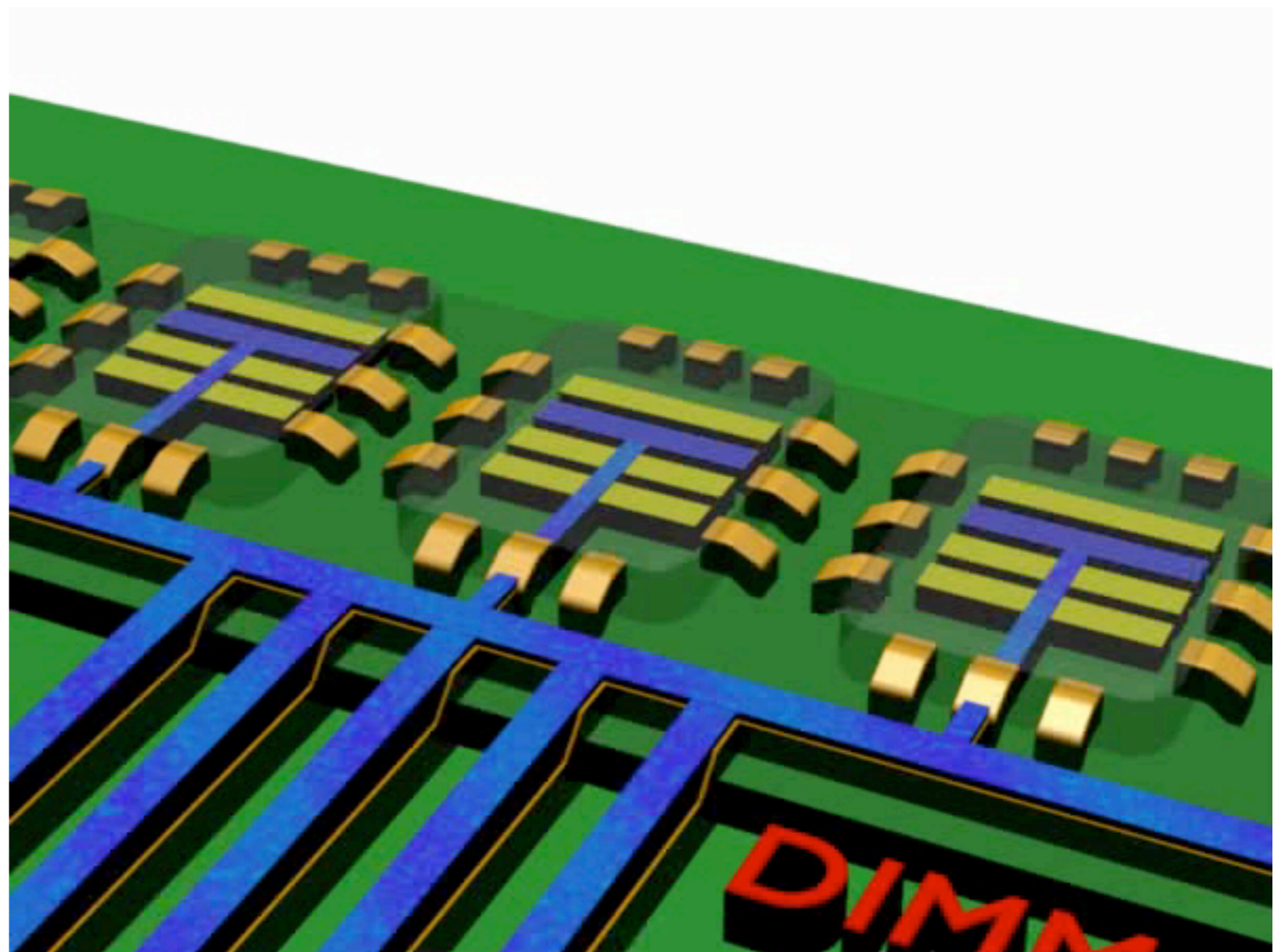
# What are we doing?

- **Two fundamental approaches to address “data movement”:**
  - **Dominant Approach: manage deep memory hierarchies (locality)**
  - **Our approach: compute near the data**
    - **Relatively shallow (5-layer) memory hierarchy (registers, scratchpad, DRAM, other cubes on a node, other cubes elsewhere)**
    - **Relaxed consistency model (move the computation, explicitly control data word state)**
    - **Memory-centric, not processor centric (memory cubes are homenodes, work initiators, control flow managers, etc.)**
- **Observation about the dominant approach**
  - **The best data locality manager today is MPI, and it does a poor job at addressing graph and stream (informatics) problems**
  - **We know we can’t manage the locality of a meaningful graph problem because it doesn’t really exist. Period.**

# Memory is efficient

## Memory Access is Inefficient

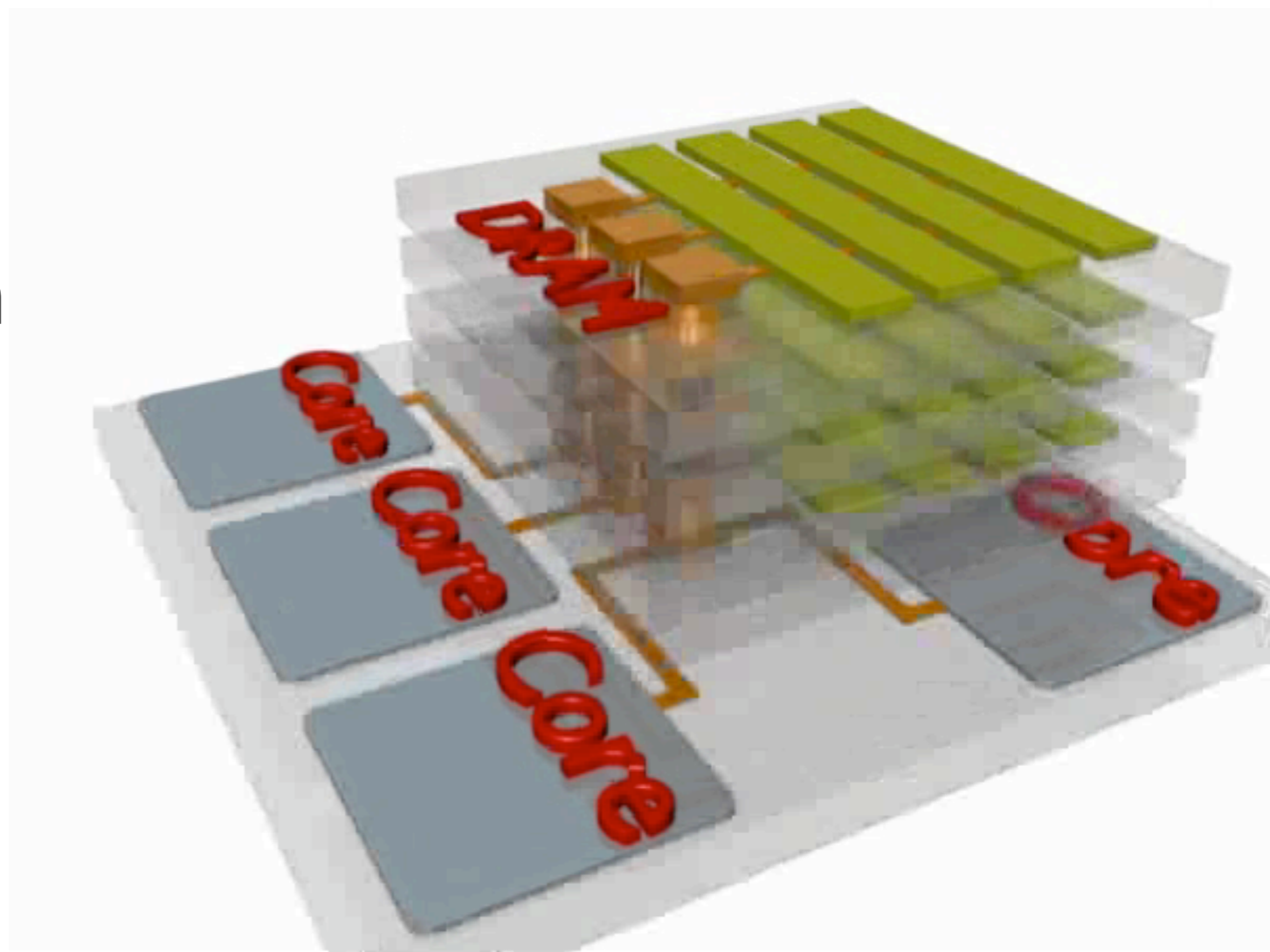
- DRAM cells require  $< 1$  pJ to access
- Current DRAM architectures are not power efficient
- Long distances  $\rightarrow$  high power
- We pay for more than we get at every level
  - Cache: throw away 75-80%
  - DRAM Row: Charge 1024B for each 64B access
  - DIMM: Charge 8-9 chips/access
  - $\sim 800$  pJ/byte total
- DRAM design driven by packaging constraints
  - $\sim 50\%$  of DRAM chip cost is packaging, mainly in pins
  - DIMMs use multiple chips with a few data pins to achieve high BW



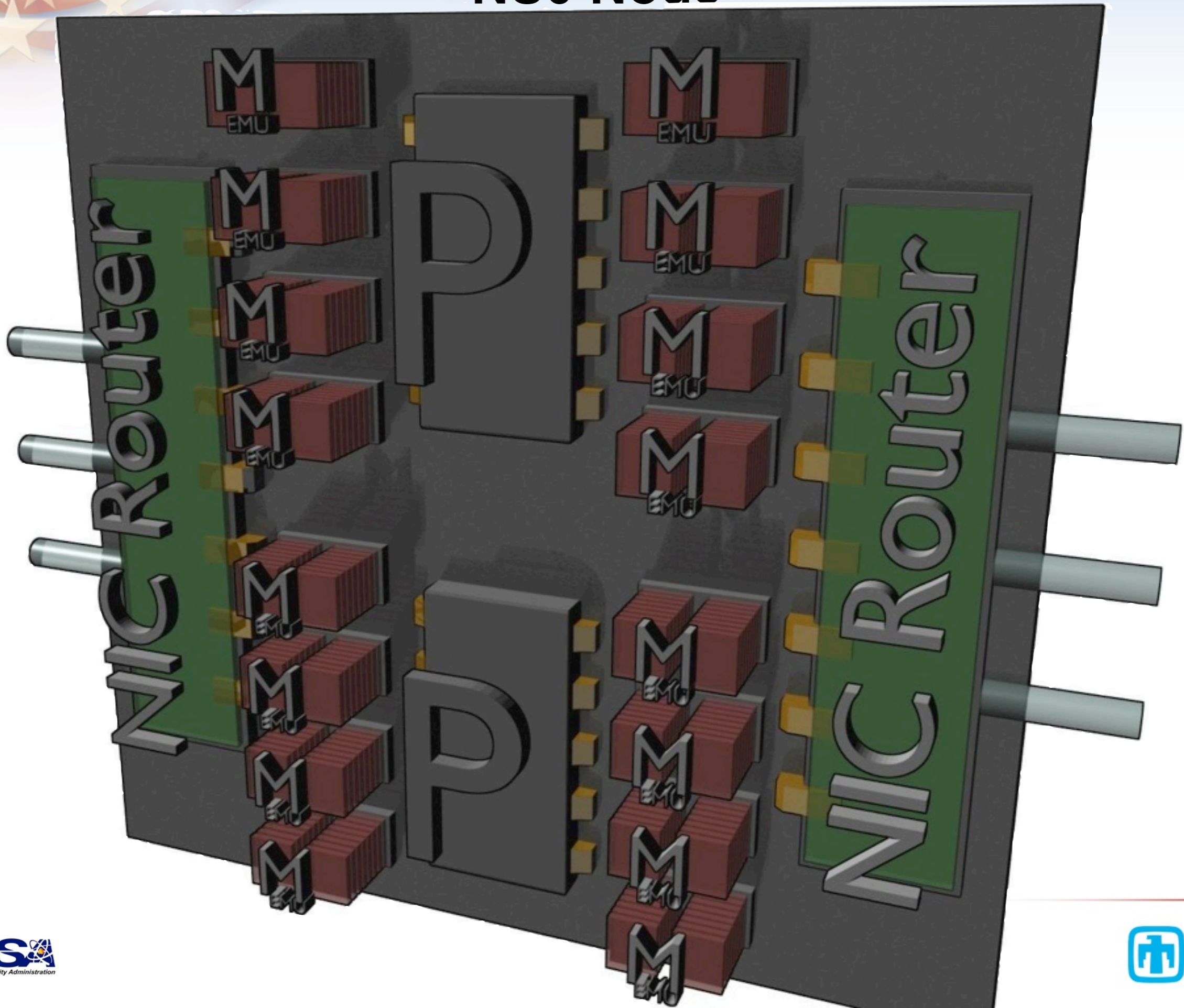
## More efficient DRAM w/ Stacking

$$Energy = (V^2 * C) * Overhead + E_{comm}$$

- Can reduce energy by reducing voltage, capacitance, overhead, or communication energy
- Voltage: Hard to scale < 1V
- Capacitance: Narrower open rows
- Overhead: Smarter memories may enable more efficient ECC
- Communication: TSVs orders of magnitude less energy
  - 2-4pJ/byte to bottom layer



# NS0 Node



# Target Scales

## • Rack Scale

– Processing: 128 Nodes, 1 (+) PF/s

– Memory:

- 128 TB DRAM

- 0.4 PB/s Aggregate Bandwidth

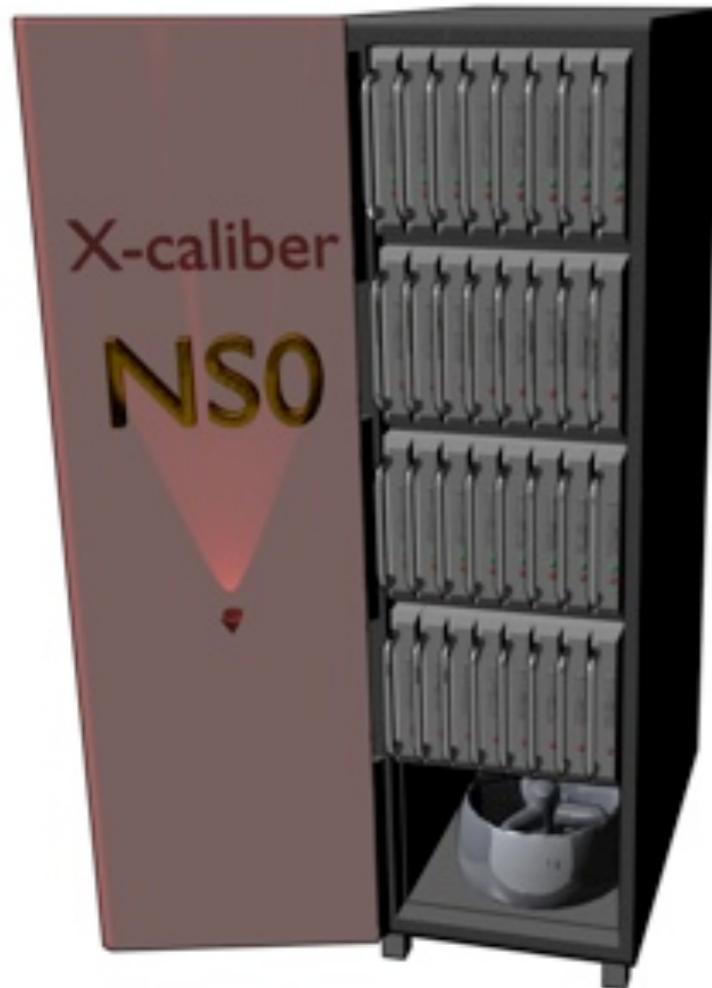
– NV Memory

- 1 PB Phase Change Memory (addressable)

- Additional 128 for Redundancy/RAID

– Network

- 0.13 PB/sec Injection, 0.06 PB/s Bisection



Deployment	Nodes	Topology	Compute	Mem BW	Injection BW	Bisection BW
Module	1	N/A	8 TF/s	3 TB/s	1 TB/s	N/A
Deployable Cage	22	All-to-All	176 TF/s	67.5 TB/s	22.5 TB/s	31 TB/s
Rack	128	Flat. Butterfly	1 PF/s	.4 PB/s	0.13 PB/s	0.066 PB/s
Group Cluster	512	Flat. Butterfly	4.1 PF/s	1.6 PB/s	0.52 PB/s	0.26 PB/s
National Resource	128k	Hier. All-to-All	1 EF/s	0.4 EB/s	0.13 EB/s	16.8 PB/s
Max Configuration	2048k	Hier. All-to-All	16 EF/s	6.4 EB/s	2.1 EB/s	0.26 EB/s

# System Balance

## • System Balance

- Because we're memory centric, we're focused on bandwidth, capacity, and scalability of the memory system (near and far)
- X-caliber compared to the state of the art (scaled to 2018):
  - 5X the FLOPs of Red Storm
  - 2X the memory capacity
  - Similar network bandwidth ratio
- Other approaches (aggregate from what I've seen):
  - 10X the FLOPs of Red Storm, Half or less the memory capacity

System	Injection BW	FLOPS	B/F	Ratio	Comment
X-caliber	133 TB/s - 266 TB/s	1.0 - 1.4 PF/s	0.095 - 0.266	1.21 - 3.38	Adaptive
Typical Exascale Thinking	205 TB/s	2.6 PF/s	0.0788	0.82 - 0.30	Static



# Execution Model

# ParalleX

Element	ParalleX Mechanism
<b>Concurrency</b>	<b>Lightweight Threads + Codelets</b> (lightweight, h/w scheduled, for latency tolerance not throughput!)
<b>Coordination</b>	<b>Lightweight Control Objects (LCOs)</b> for construction of mutexes, futures, producer/consumer interactions, etc.
<b>Movement</b>	<b>Of Work: Parcels (lightweight active messages)</b> <b>Of Data: PGAS and Bulk Transfer</b>
<b>Naming</b>	<b>Global Name Space and Global Address Space</b>
<b>Introspection</b>	<b>Unified publication at all levels via System Knowledge Graph (SKG)</b>

# ParalleX vs. Other Models

Element	ParalleX	GPUs	Stylized CSP	PGAS
<b>Concurrency</b>	<b>Threads/Codelets</b>	<b>SIMD/lock-step threads</b>	<b>Ranks/ Processes</b>	<b>Processes</b>
<b>Coordination</b>	<b>Lightweight Control Objects (fine-grained)</b>	<b>Local Memory/ Explicit</b>	<b>BSP</b>	<b>BSP</b>
<b>Movement</b>	<b>of Work: Parcels of Data: PGAS and Bulk</b>	<b>Bulk Data Transfer (weak memory system)</b>	<b>Bulk Data Transfer</b>	<b>Data Only (load + store)</b>
<b>Naming</b>	<b>Global Name Space Global Address Space</b>	<b>Global Address Space</b>	<b>Explicit by Rank</b>	<b>Global Address Space</b>
<b>Introspection and Adaptivity</b>	<b>System Knowledge Graph/Dynamic</b>	<b>None/Static</b>	<b>None/Static</b>	<b>None/Static</b>

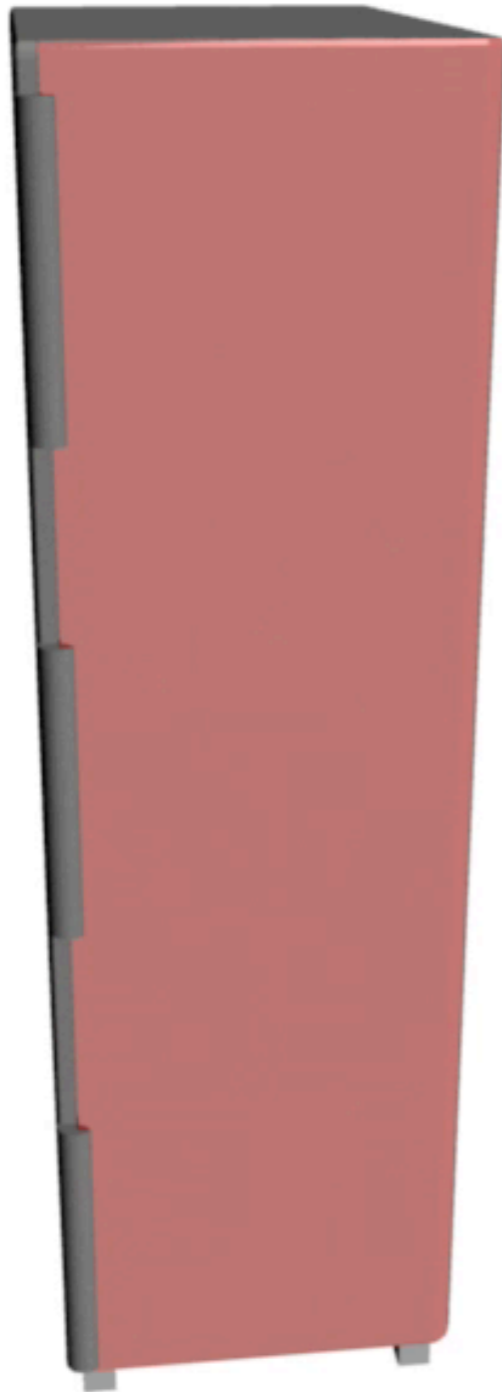
# X-Caliber

- **Focus Areas**

- Technology
- Architecture
- Execution Models

- **Cross cutting themes**

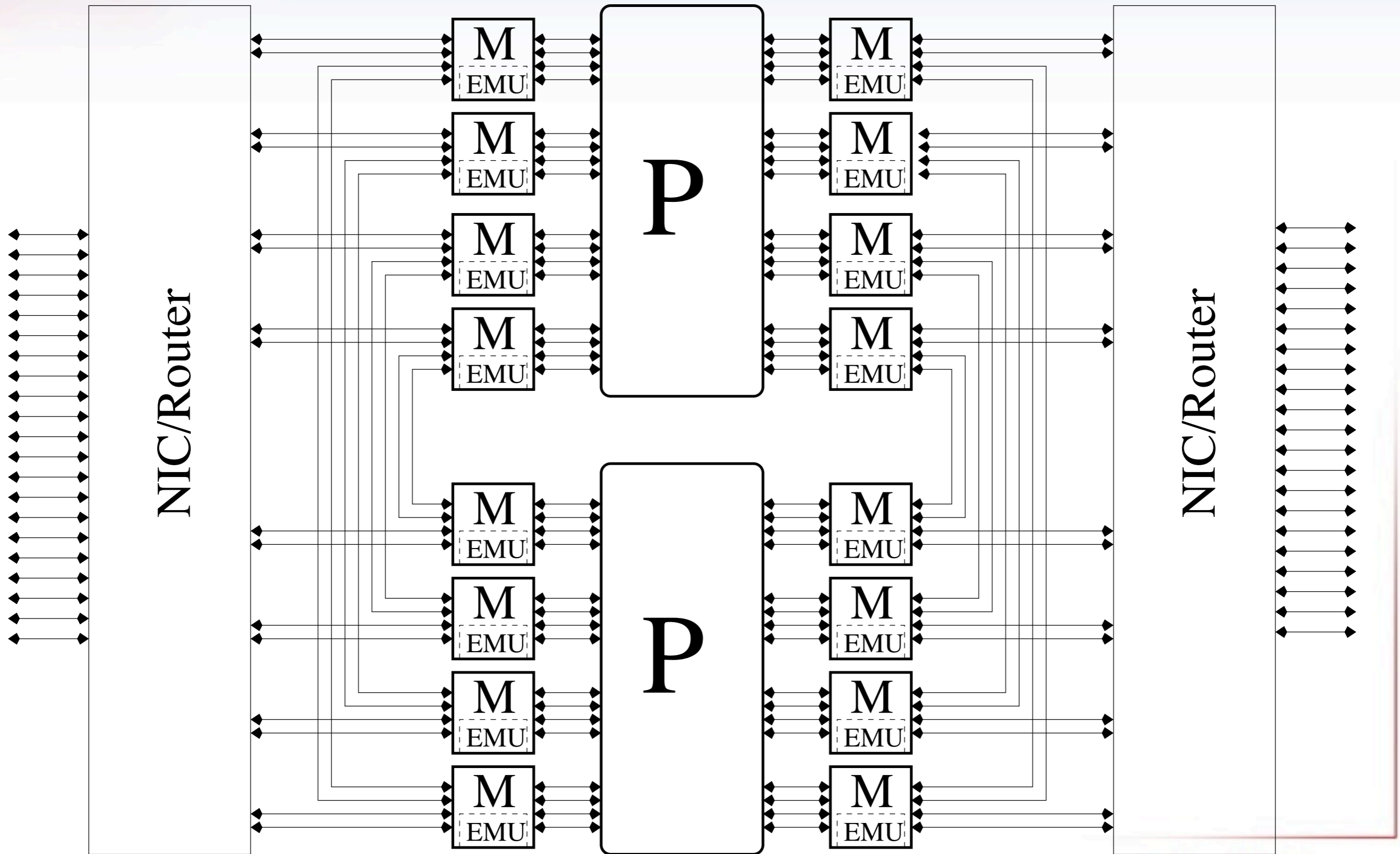
- Data movement is costly
- Power is critical





**Thank You**

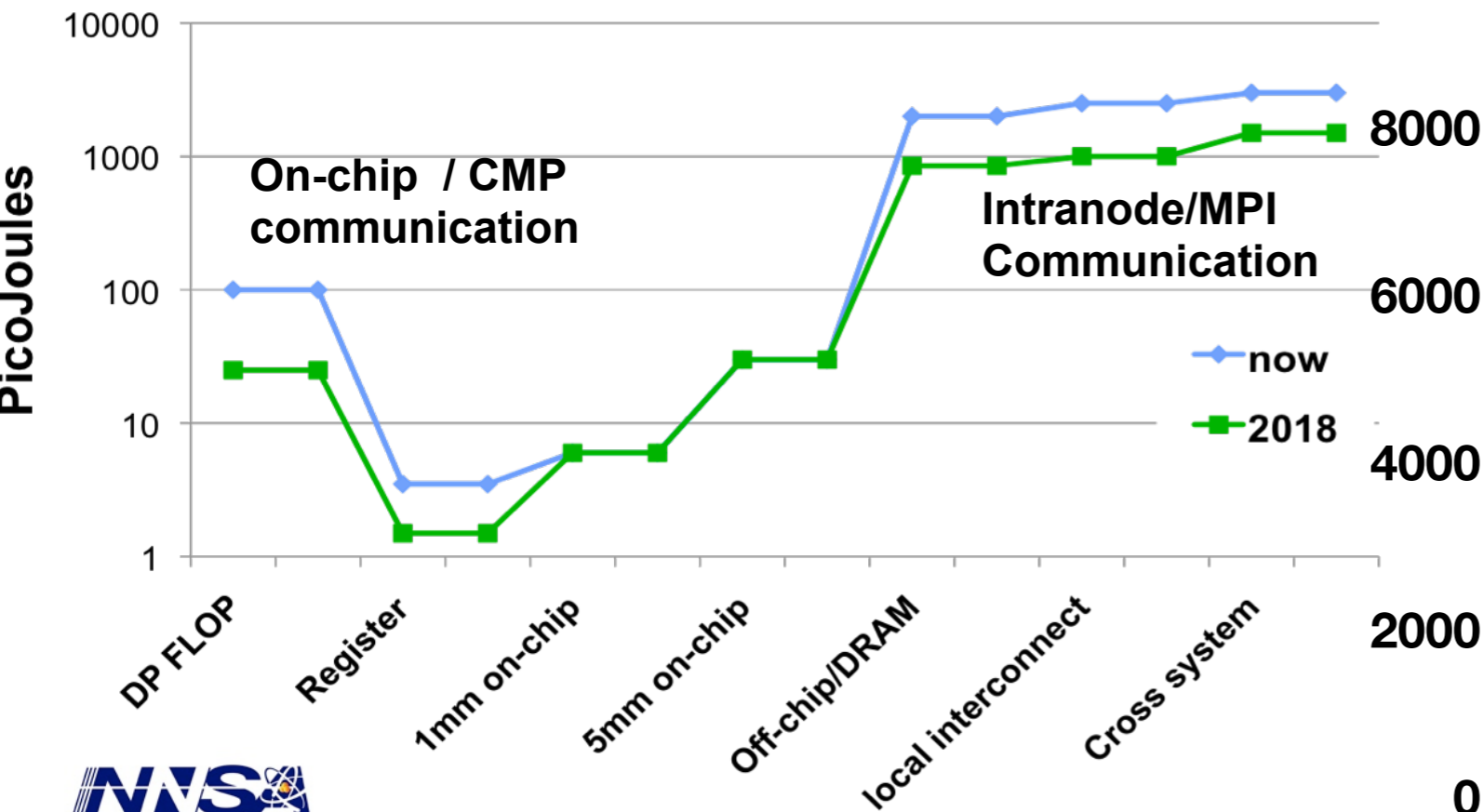
# Node Architecture (Continued)



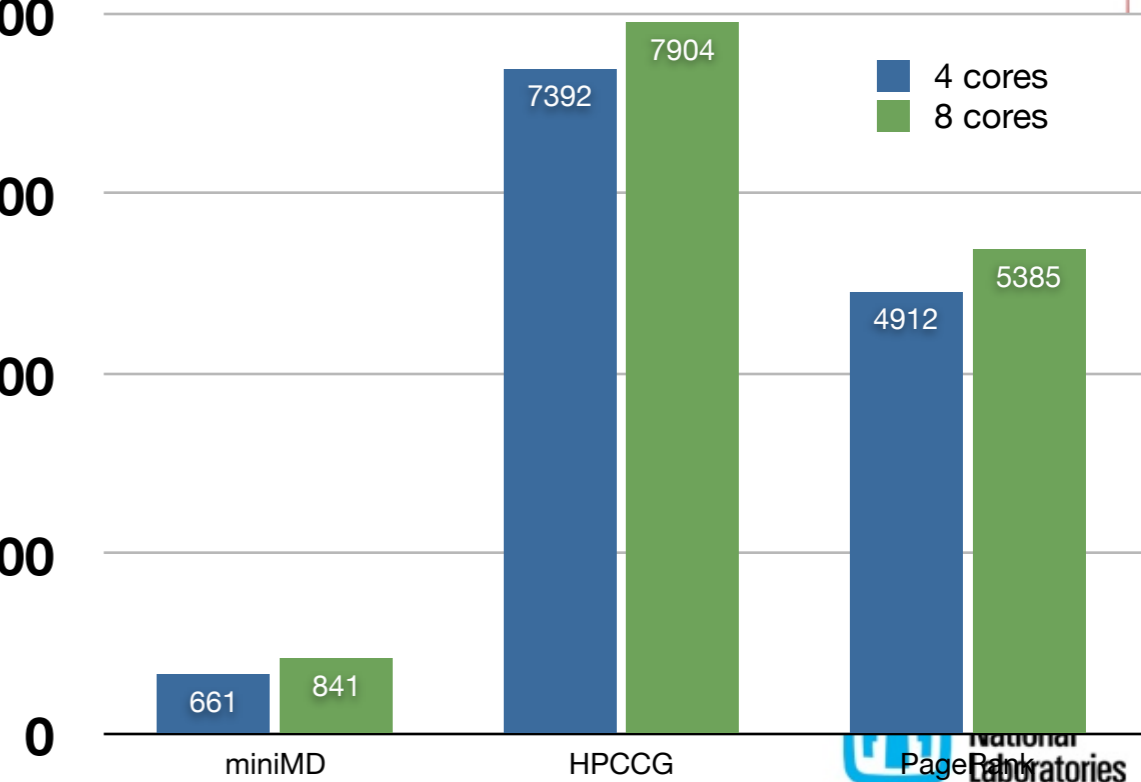
# Data Movement Dominates Power

- 10-12pJ/FLOP in 2018
- ~4-8 pJ/word to access registers
- 100s pJ/byte to access DRAM
- 1000s pJ to move across system
- With L2, NoC, memory controllers, & DRAM applications typically consume several hundred to several thousand times more energy in the memory system than in the floating point unit

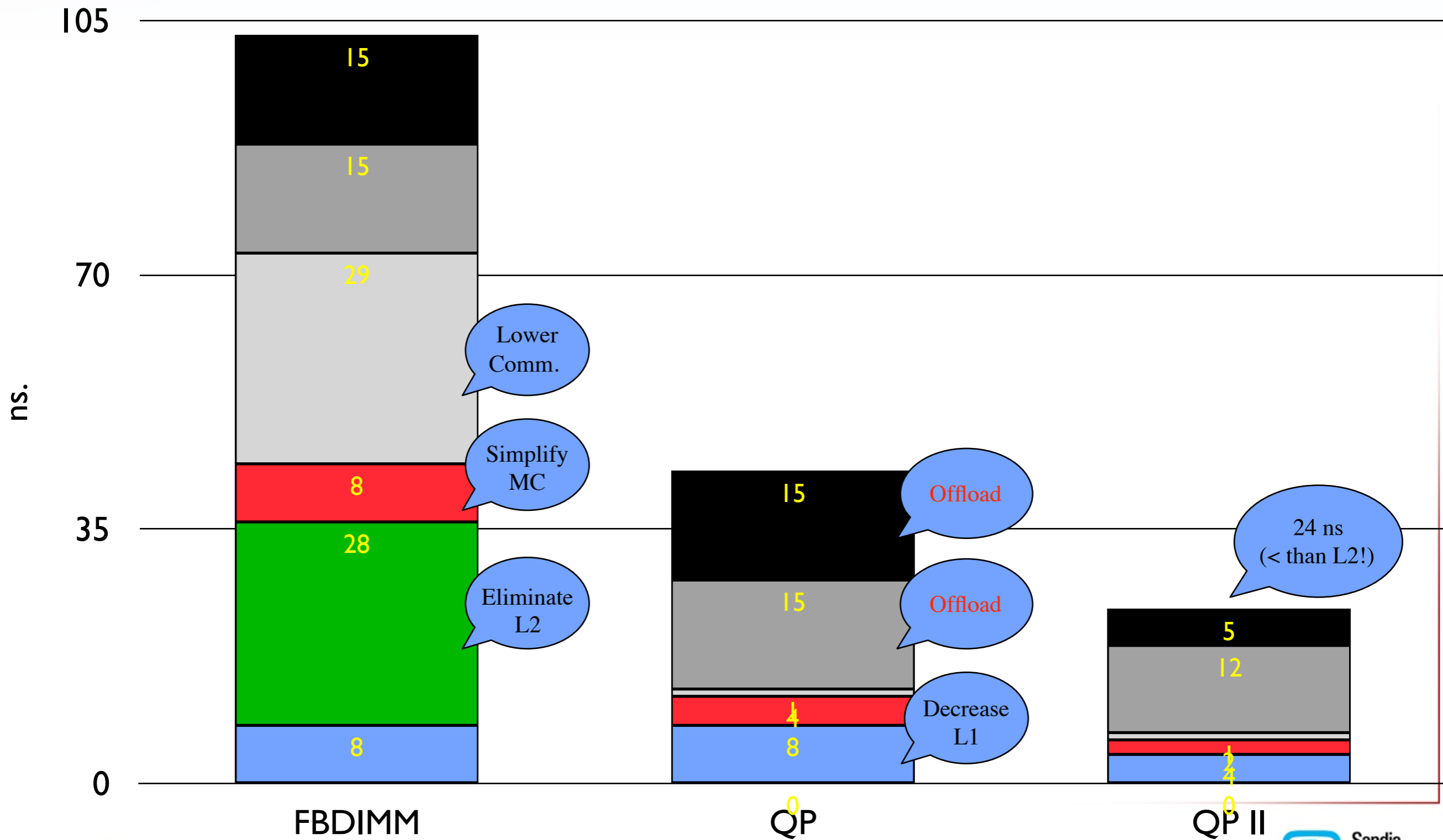
“The Energy and Power Challenge is the most pervasive ... and has its roots in the inability of the [study] group to project any combination of currently mature technologies that will deliver sufficiently powerful systems in any class at the desired levels.”  
*DARPA IPTO exascale technology challenge report*



Ratio of Memory Energy to FPU Energy



# Performance Potential: Latency Reduction



# External Advisory Board and Systems Integrator Panel

- **External Advisory Board**

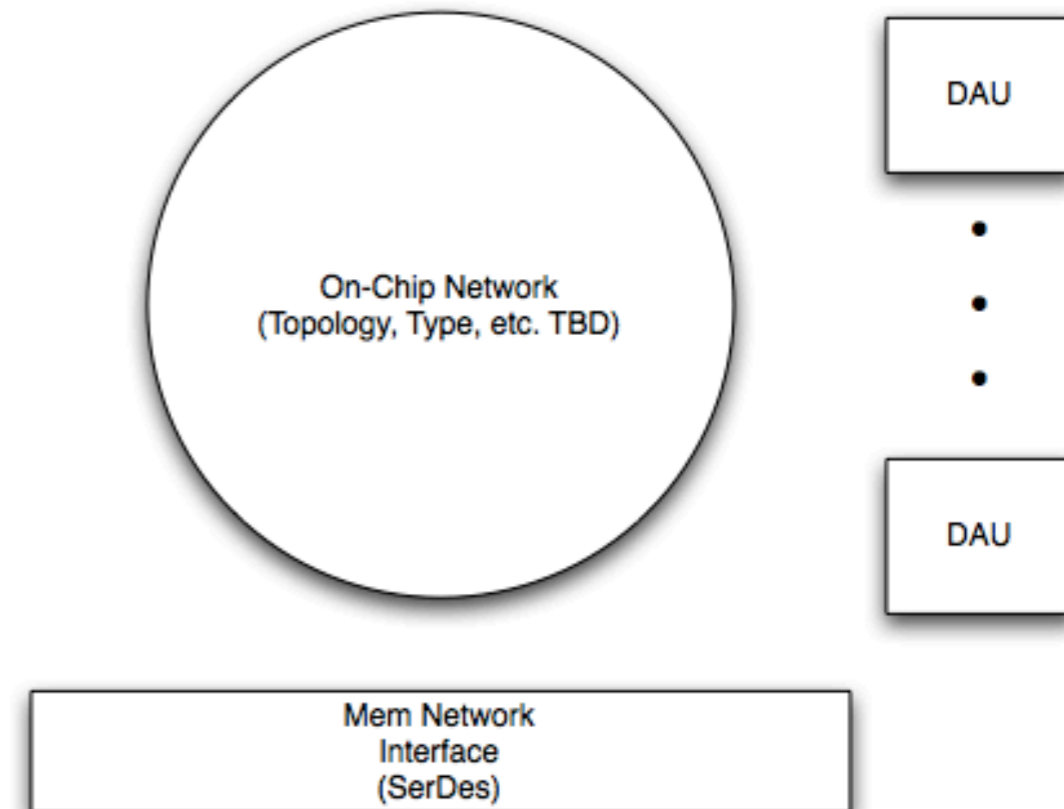
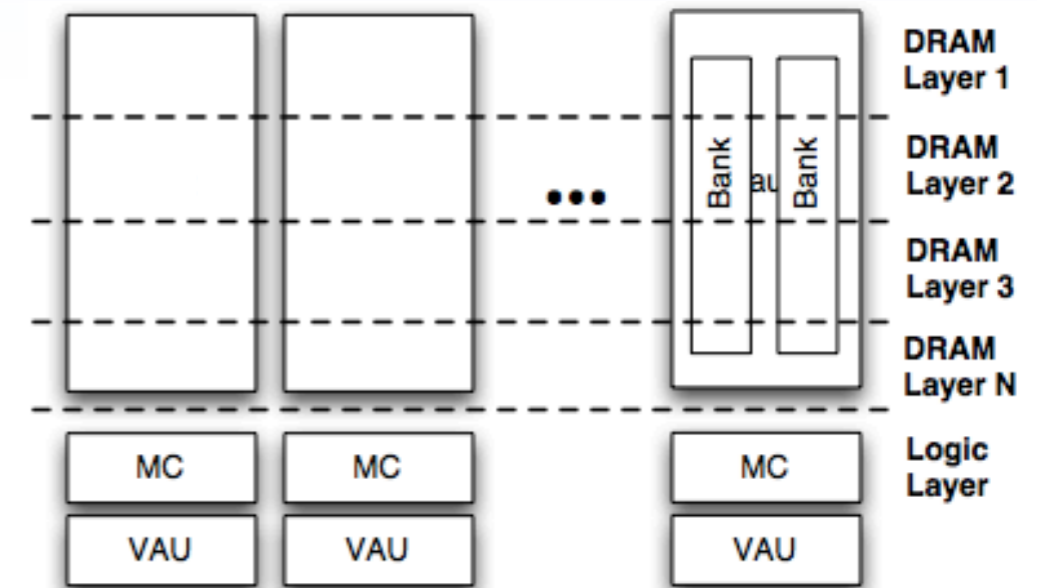
- Larry Bergman, JPL
- Mike Levine, PSC
- John Morrison, LANL
- Jeff Nichols, ORNL
- Dan Reed, Microsoft
- Sander Lee, NNSA
- Dave Mountain, DoD
- Steve Poole

- **Systems Integrator Panel**

- Help understand the tera-scale environment and non-supercomputer deployments
- Rockwell Collins, Lockheed Martin, Northrop Grumman, GE Research, Raytheon, Boeing, QUALCOMM

# The EMP

- 3D Stack: DRAM & Logic
- Memory Controllers (read/write → RAS/CAS)
- On Chip network
- Off-chip communication
- Multiple Processing elements
  - ‘DAUs’: Close to memory controller
  - ‘VAUs’: Closer
- Design space:
  - # of VAUs/DAUs/MCs, bandwidths, topologies, etc...



# Where are we departing from the roadmap?

Element	ParalleX	GPUs	Stylized CSP	PGAS
Concurrency	<b>Threads/Codelets</b>	<b>SIMD/lock-step threads</b>	<b>Ranks/ Processes</b>	<b>Processes</b>
Coordination	<b>Lightweight Control Objects (fine-grained)</b>	<b>Local Memory/ Explicit</b>	<b>BSP</b>	<b>BSP</b>
Movement	<b>of Work: Parcels of Data: PGAS + Bulk</b>	<b>Bulk Data Transfer (weak memory system)</b>	<b>Bulk Data Transfer</b>	<b>Data Only (load + store)</b>
Naming	<b>Global Name Space</b> Global Address Space	<b>Global Address Space</b>	<b>Explicit by Rank</b>	<b>Global Address Space</b>
Introspection and Adaptivity	<b>System Knowledge Graph/Dynamic</b>	<b>None/Static</b>	<b>None/Static</b>	<b>None/Static</b>

# Inside the DAUs

- **Simple pipeline of some sort**
  - Wide access(?)
  - Multithreaded
- **Memory/NoC access**
- **Interesting bits...**
  - Scratchpad: vs cache. Shared w/ registers? globally addressable?
  - Instruction encoding: Compressed? Contains dataflow state?
  - Global address space
  - Integration w/ network (parcel handling)
  - Integration w/ memory

