Reading the ~~Tea-Leaves:~~ Entrails: How Architecture Has Evolved at the High End

Peter M. Kogge
McCourtney Prof. of CS & Engr.
University of Notre Dame

Acknowledgement: This work was funded in part by the US Dept. of Energy, Sandia National Labs, as part of their XGC project, and in part by the DOE PSAAP C-SWARM Project.

Rationale Behind This Talk

- Track technology trends
- Understand "best of breed"
- Project ahead
- Identify "tall poles"
- Do so on yearly basis
  - First as part of EXASCALE technology report
  - Then as yearly updates under Sandia XGC project
  - In future under DOE PSAAP C-SWARM Center

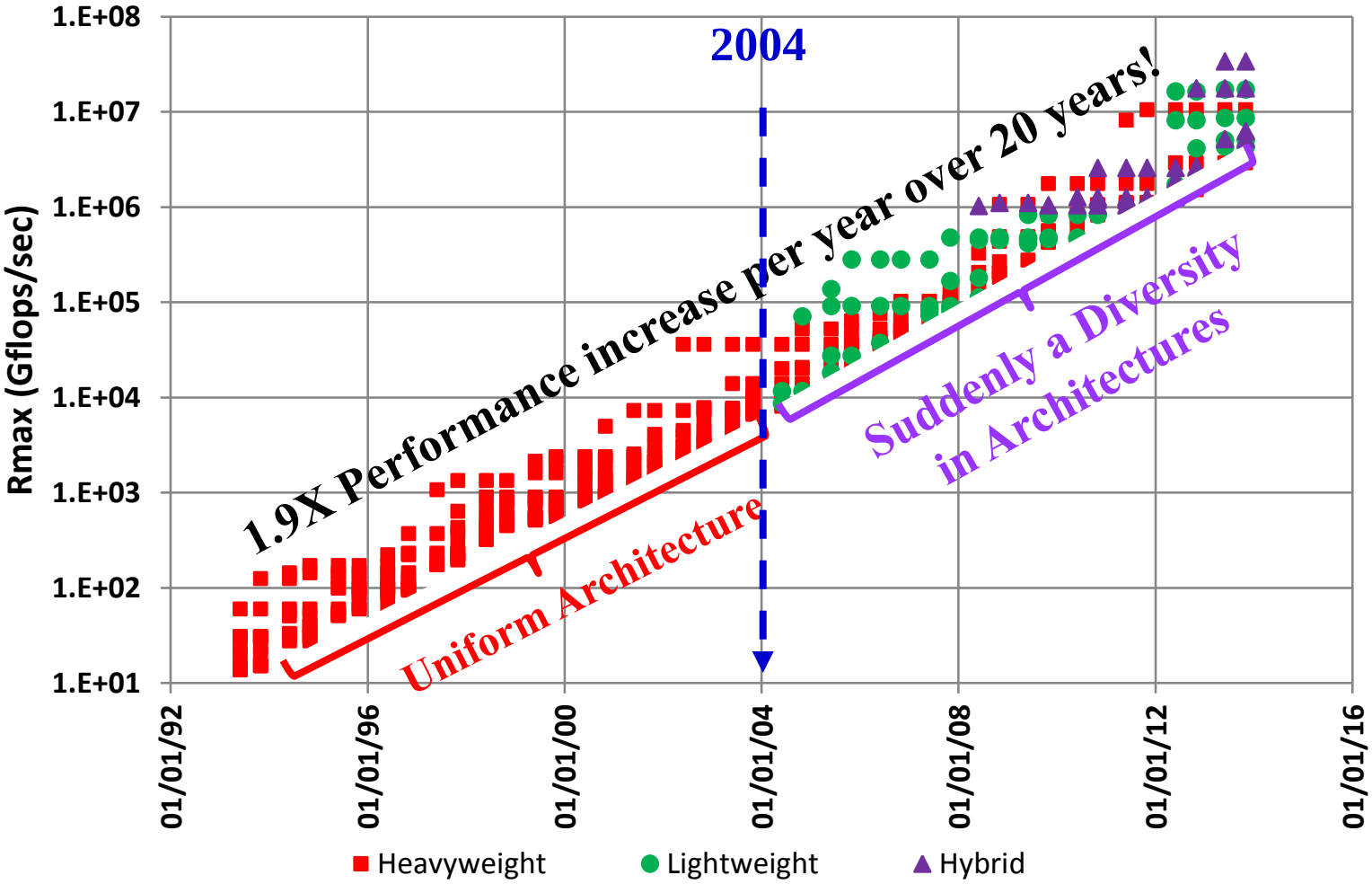
Topics

- **2004: Technology "Perfect Storm"**
- Our Current Architecture Spectrum
- The Exascale Memory Energy Horror Show
- TOP500 Lessons
- GRAPH500 Lessons
- HPCC Lessons
- 3D Stack Projections

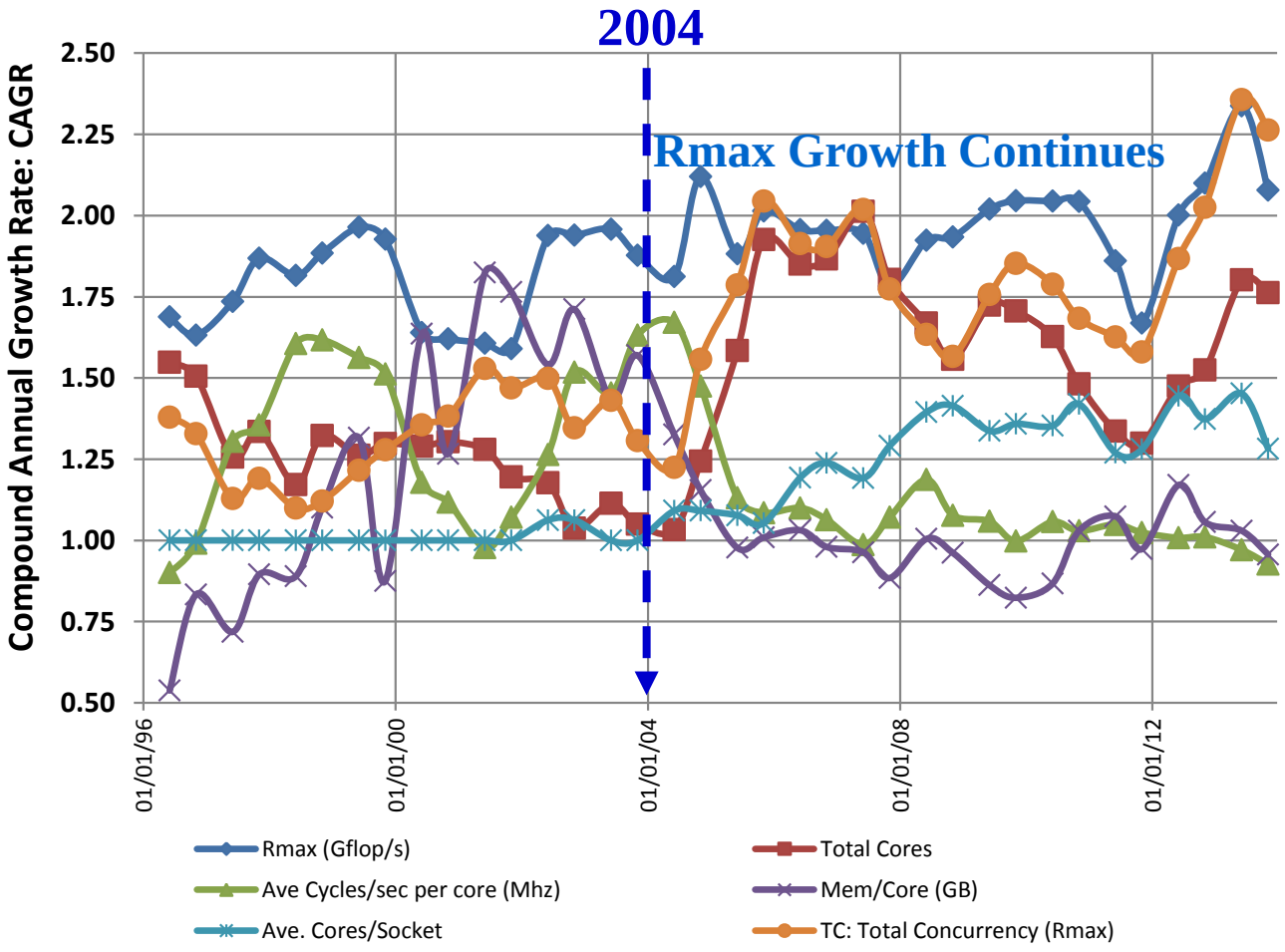
The Technology Environment

- We all believe death of Moore's Law is in sight
- And "TOP500" has continued unabated for 20 years
- But the world already changed in 2004
- Problem: Power Wall Limits Clock
- Problem: Loss of Memory Density
- Problem: Loss of off Chip Bandwidth
- Result: Have entered era of very massive parallelism

We All Know The Story: Unbroken Growth in TOP500 Rmax



But When We Look Deeper Something *Big* Happened in 2004!



Flops/cycle increased even faster

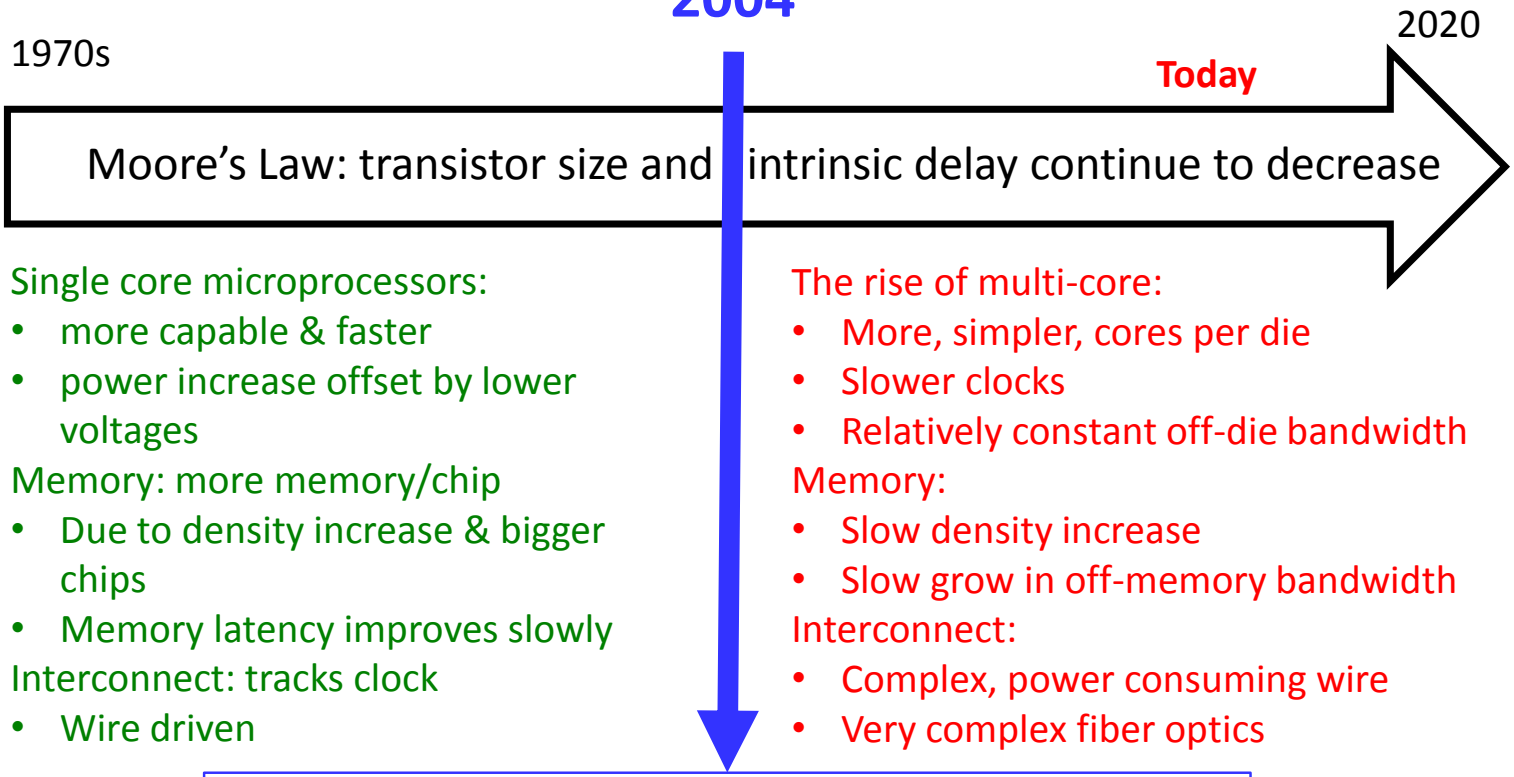
Total Cores increased even faster

Cores/socket increased

Memory/core went flat

Clock rates went flat

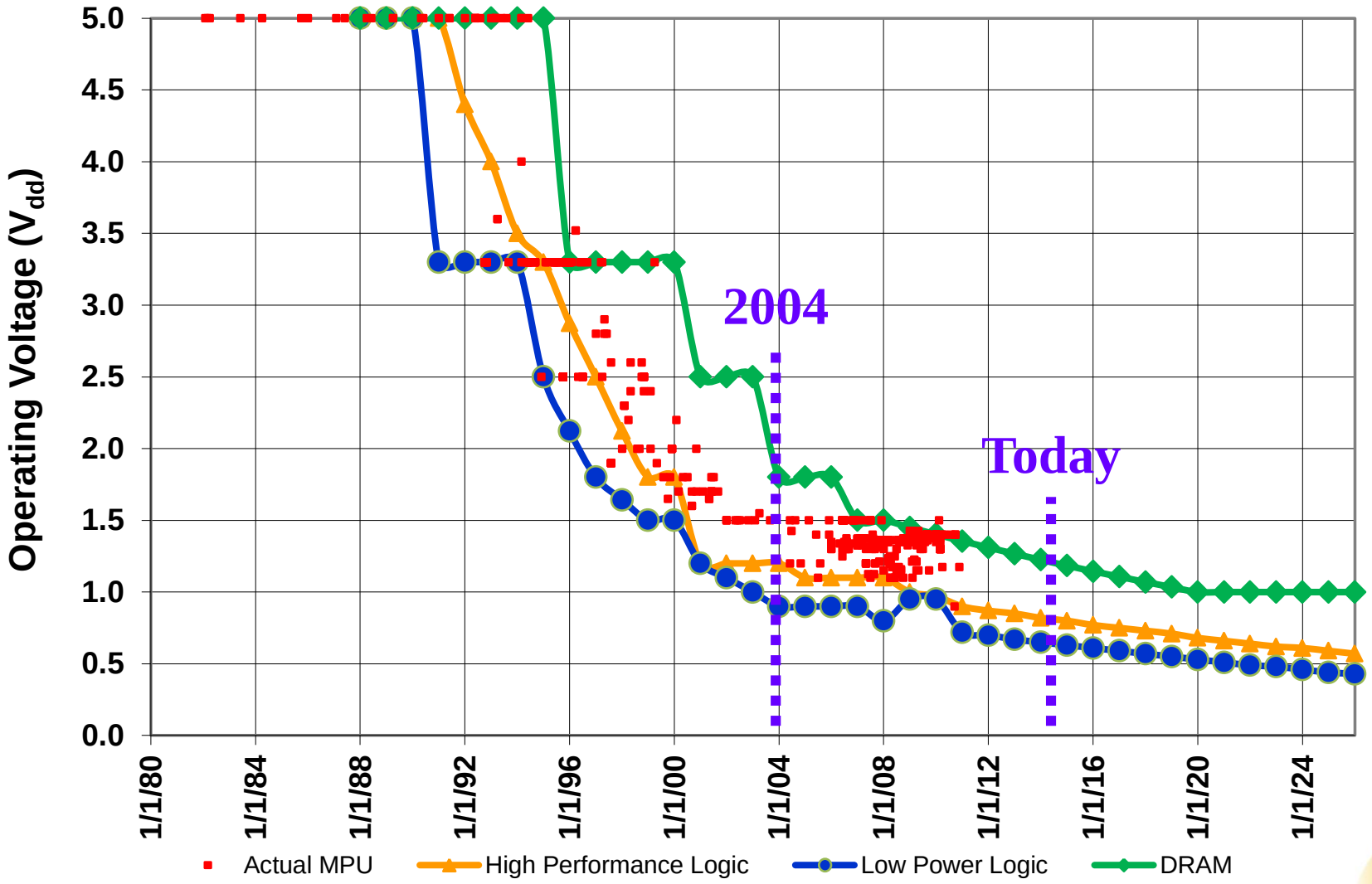
The 2004 "Perfect Storm"



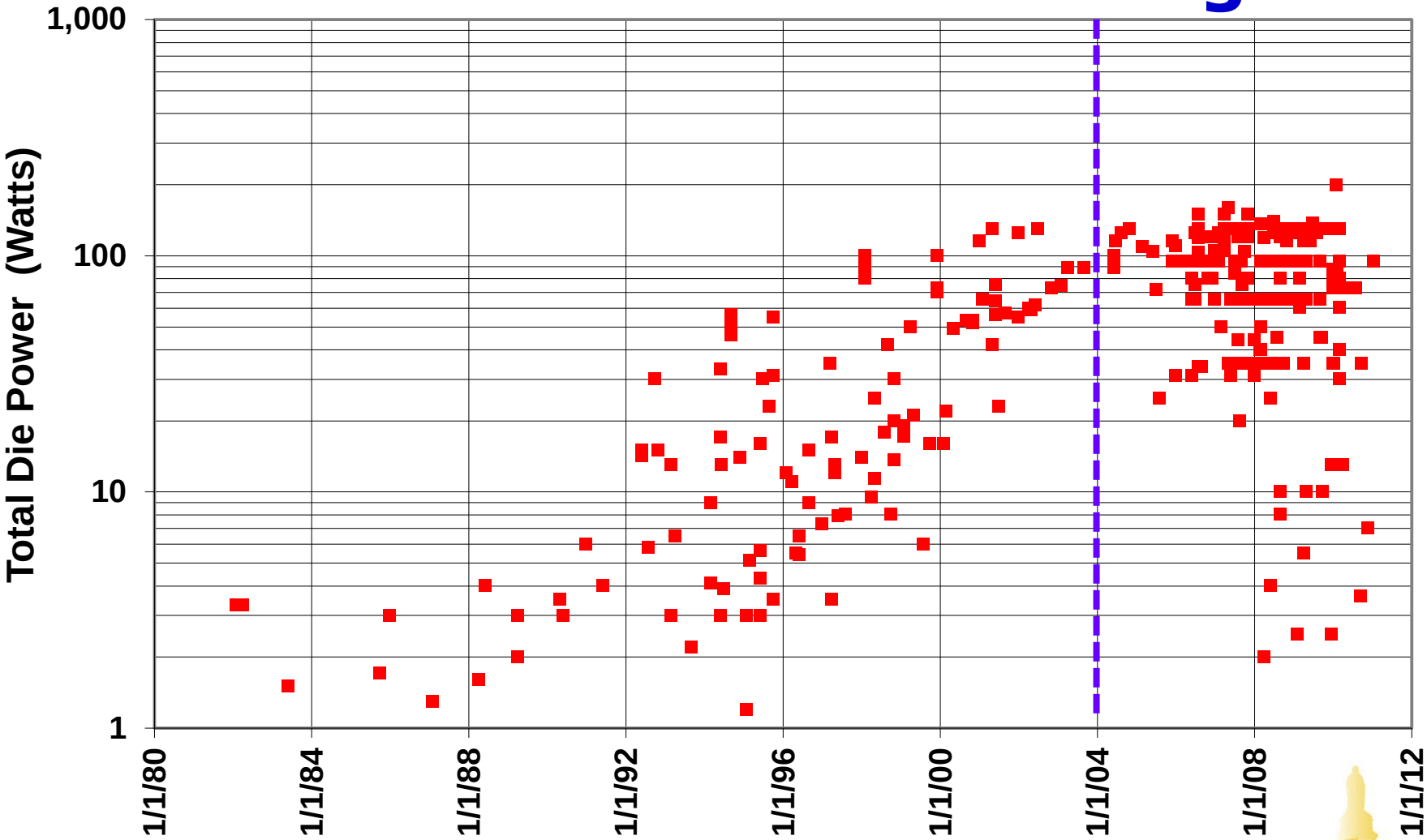
Power: *THE* 1st Design Constraint

- Moore's Law: feature reduction by "S" per year
- Power in CMOS circuit: αCFV^2
  - αC: average capacitance switched per clock cycle; Scales ≈1/S
  - F: clock rate; Intrinsic peak scales as S
  - V: operating voltage: This is the problem!
- αCF essentially flat for different Moore generations
- But we can pack S^2 more circuits on same chip size
- Thus power: $S^2 V^2$
- Before 2004, V declined, reducing increase in power
- After 2004, V has flattened; power can *run wild*!
  - And better cooling is too expensive
- Only option: reduce clock

Flattening Operating Voltage

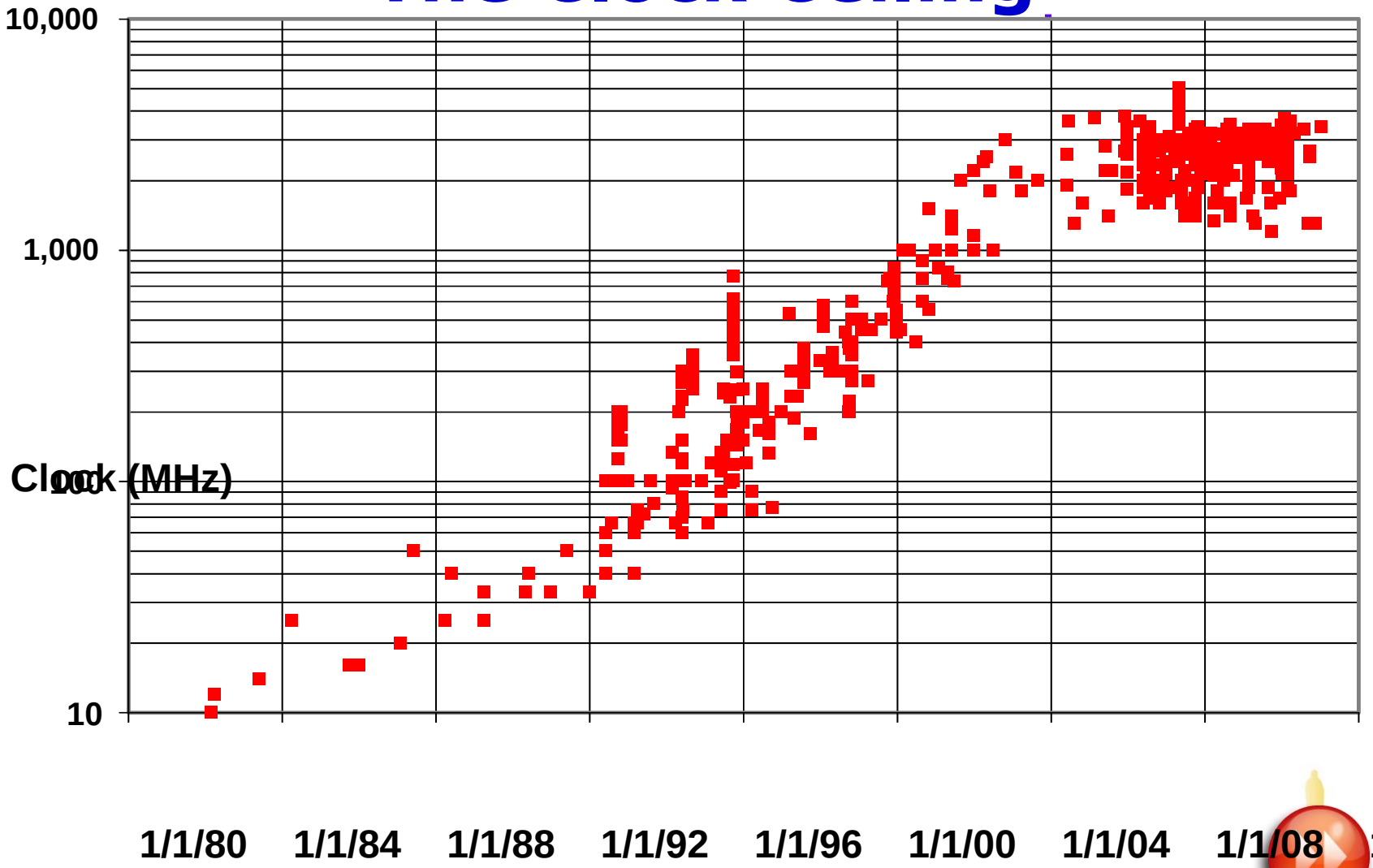


We've Hit a Power Ceiling



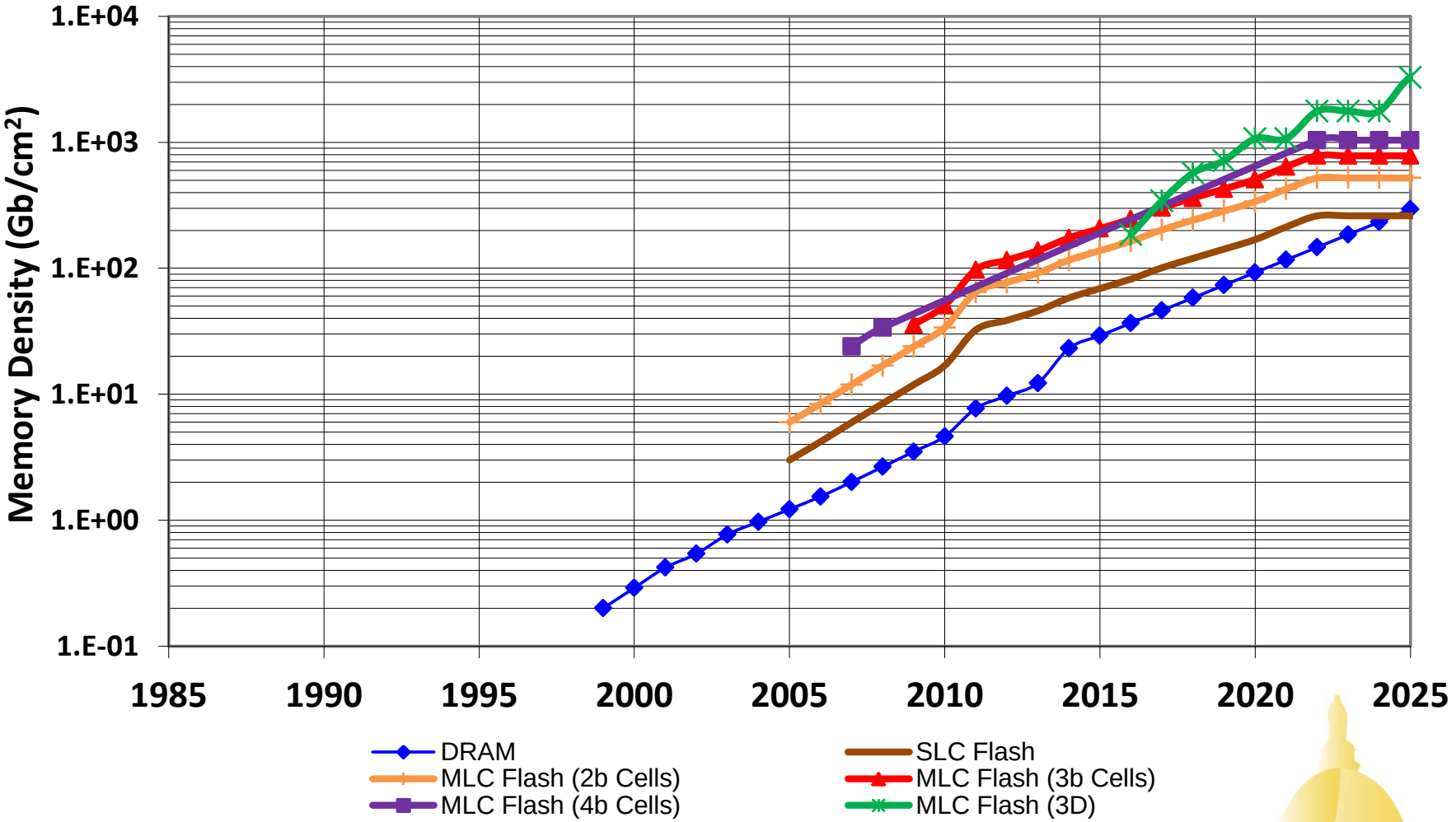
data from www.cpudb.stanford.edu

The Clock Ceiling

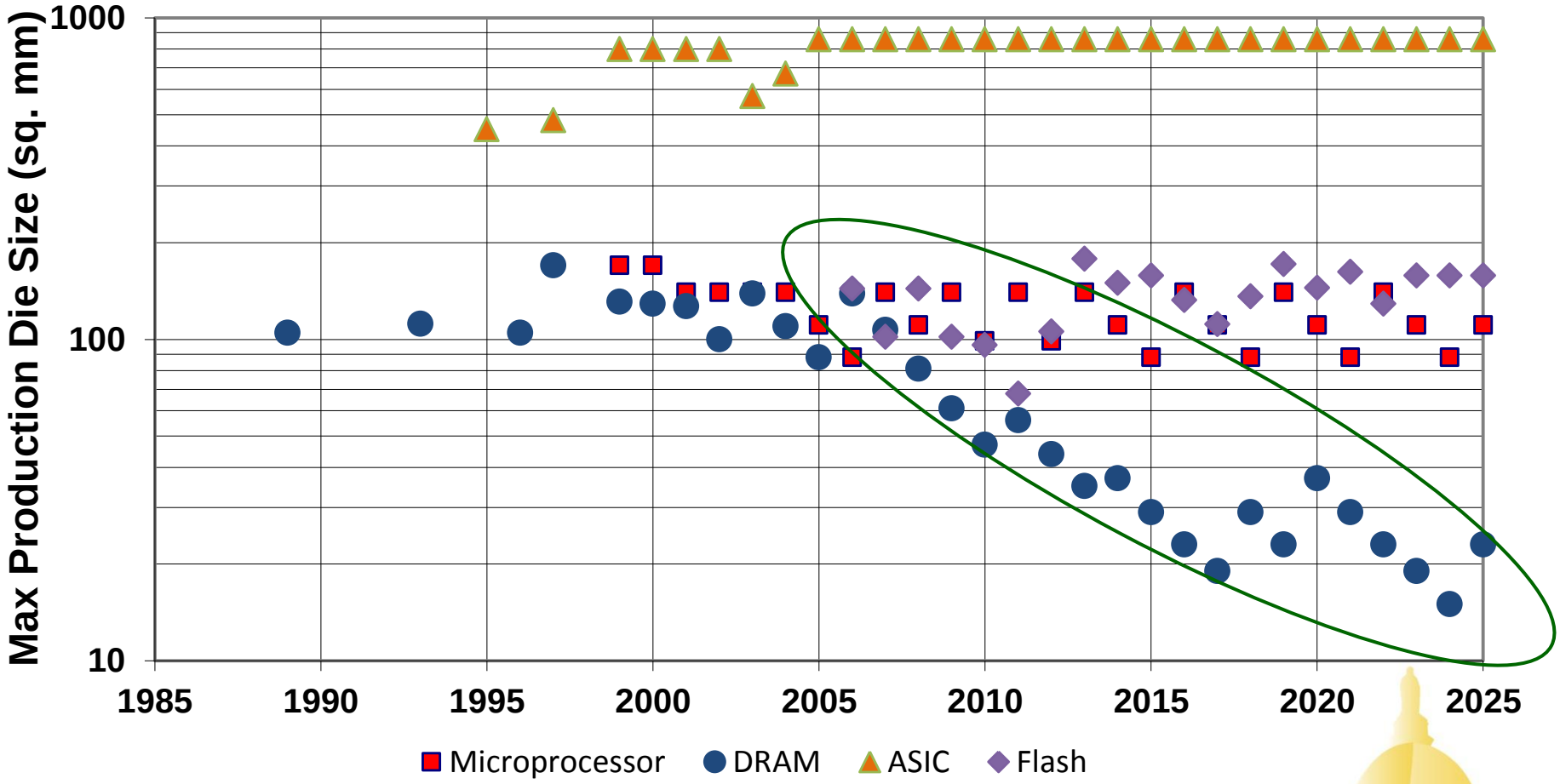


data from www.cpudb.stanford.edu

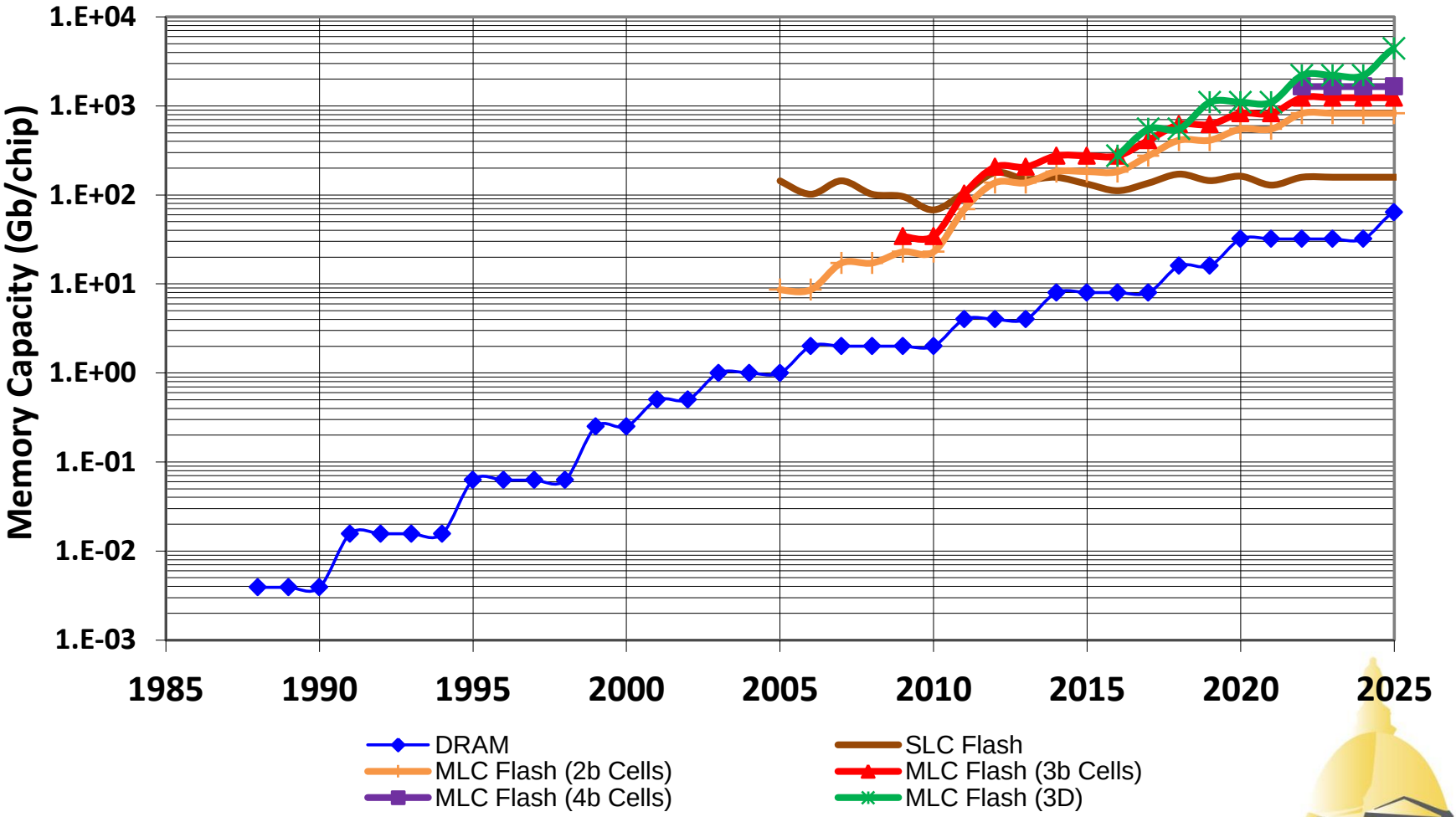
Memory Density Increasing, But



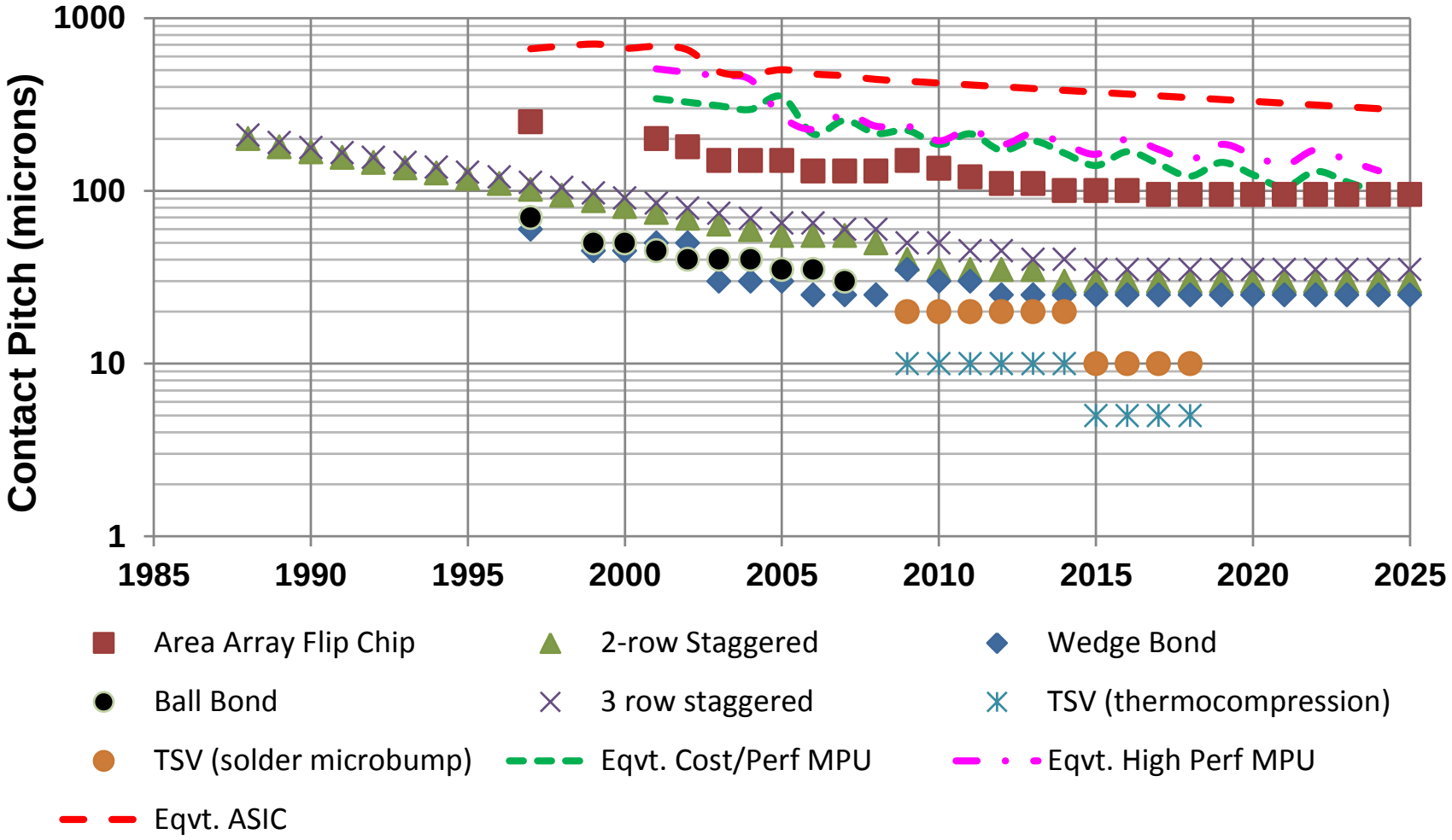
DRAM Die Sizes Are Flattening or Decreasing



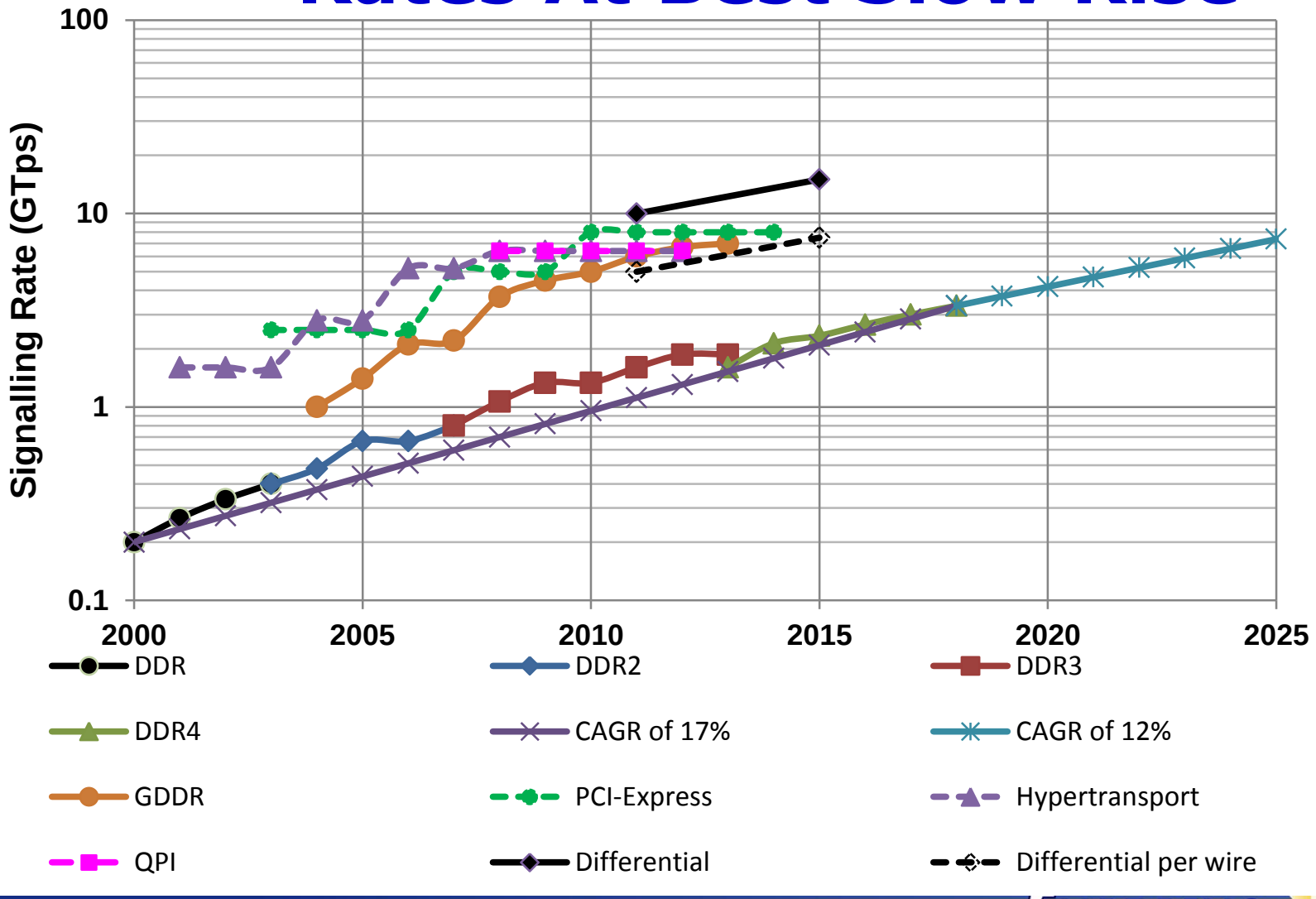
So Memory Density Growth/Die is Slowing



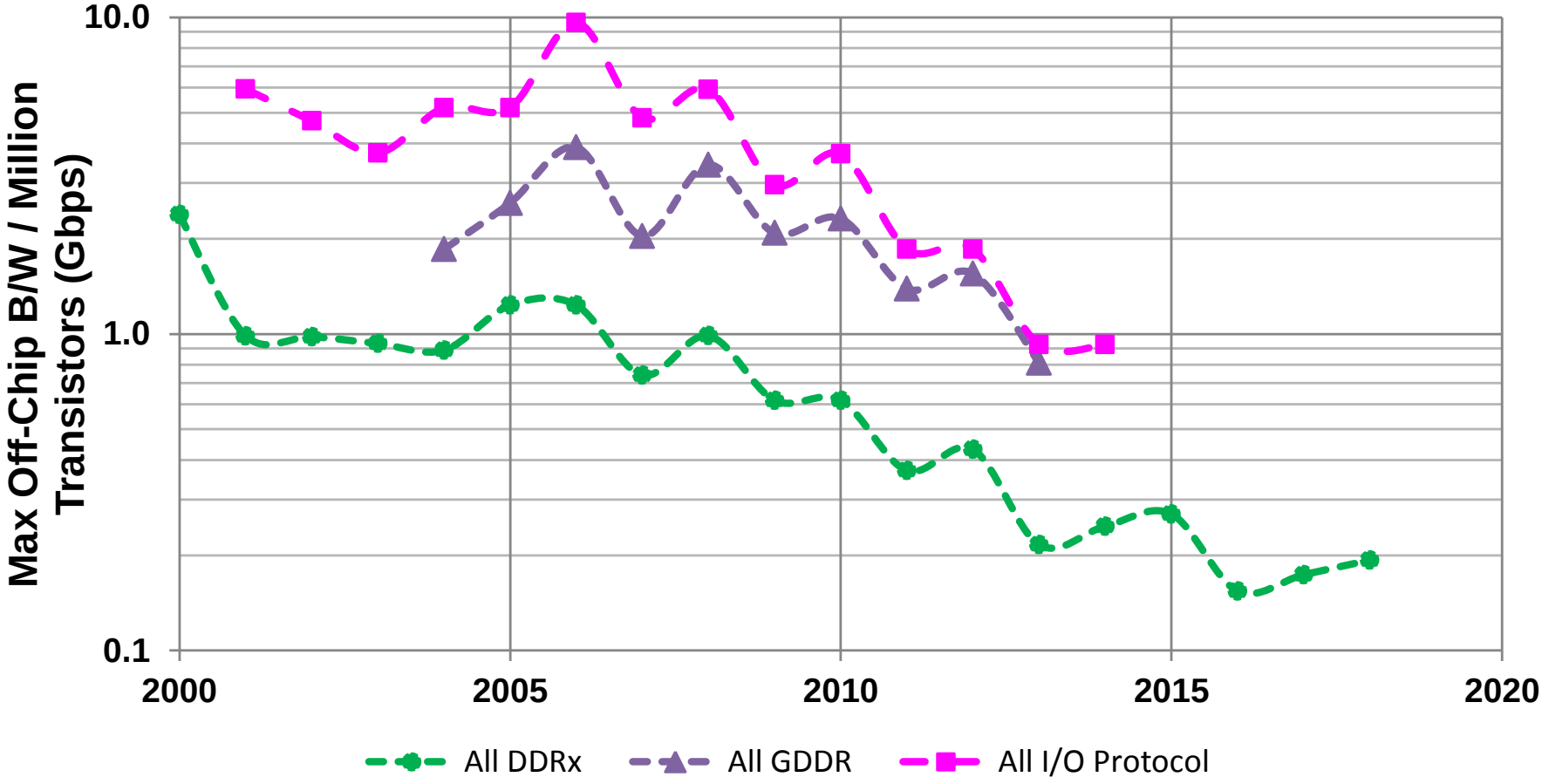
Chip I/O Pitch is Flattening



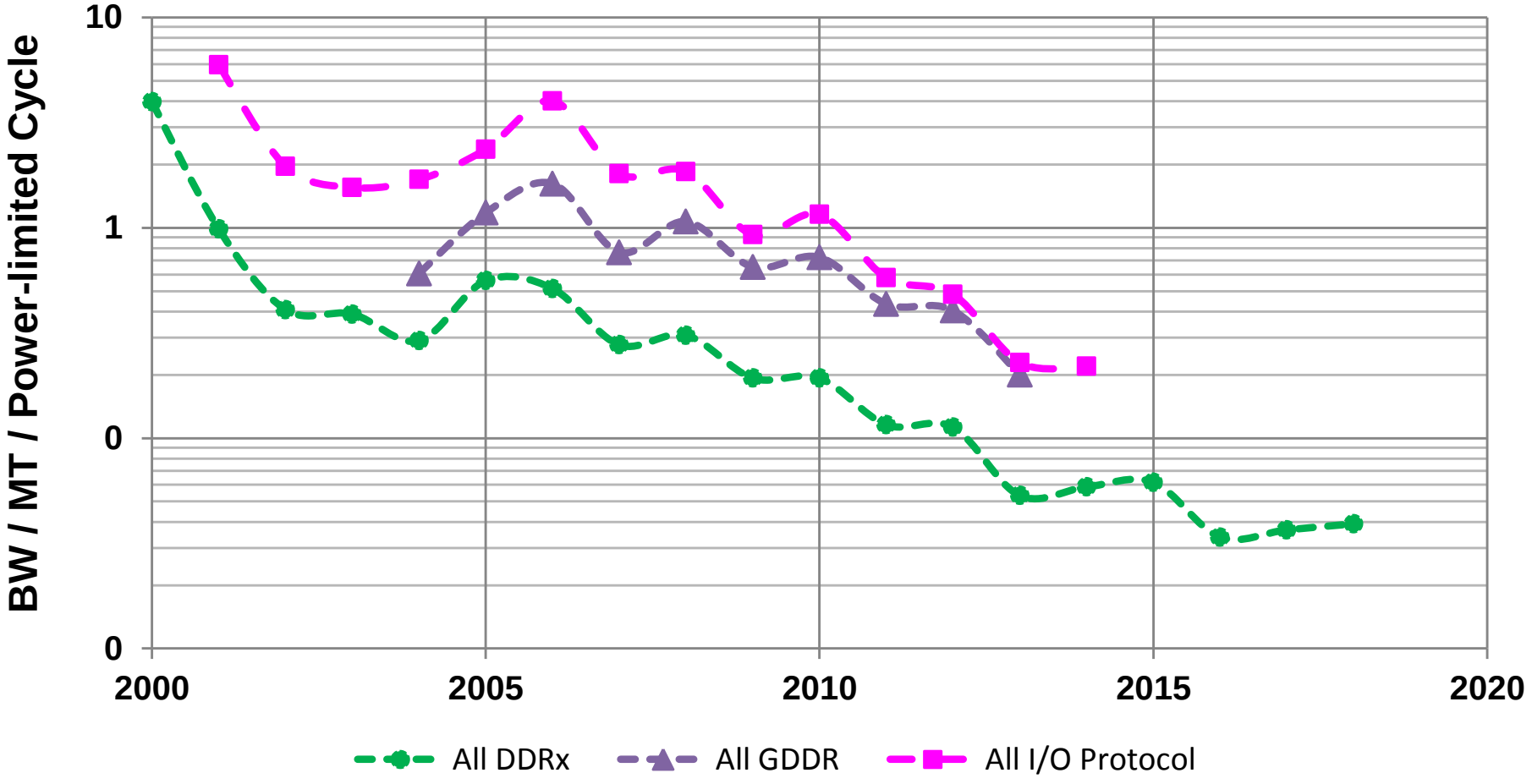
And Commodity Off-Chip Signaling Rates At Best Slow Rise



Thus "Per Unit Logic" Off-Chip B/W Decaying



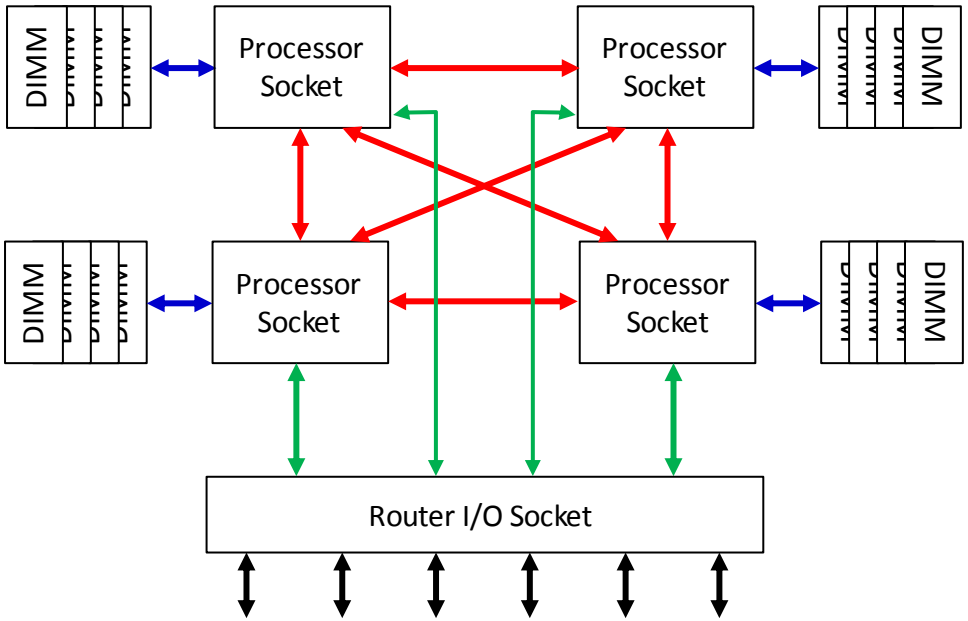
With Even Less B/W When We Normalize to "per Cycle"



Topics

- 2004: Technology "Perfect Storm"
- **Our Current Architecture Spectrum**
- The Exascale Memory Energy Horror Show
- TOP500 Lessons
- GRAPH500 Lessons
- HPCC Lessons
- 3D Stack Projections

Today's Heavyweight

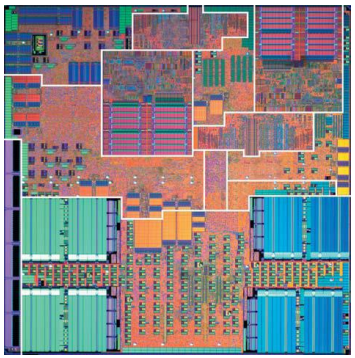


A Power 7 Drawer

LightWeight Architectures Starting with BlueGene L

2 Nodes per "Compute Card." Each node:
- **A low power compute chip**
- **Some memory chips**
- **"Nothing Else"**

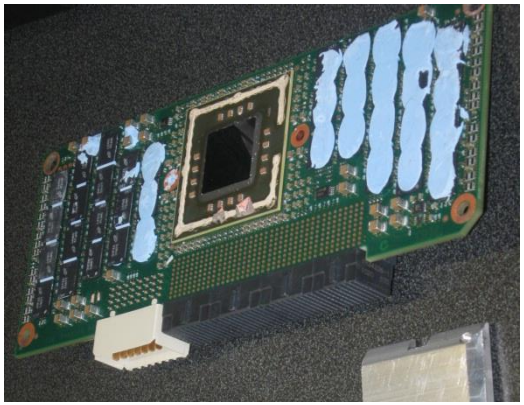
System Architecture:
- **Multiple Identical Boards/Rack**
- **Each board holds multiple Compute Cards**
- **"Nothing Else"**

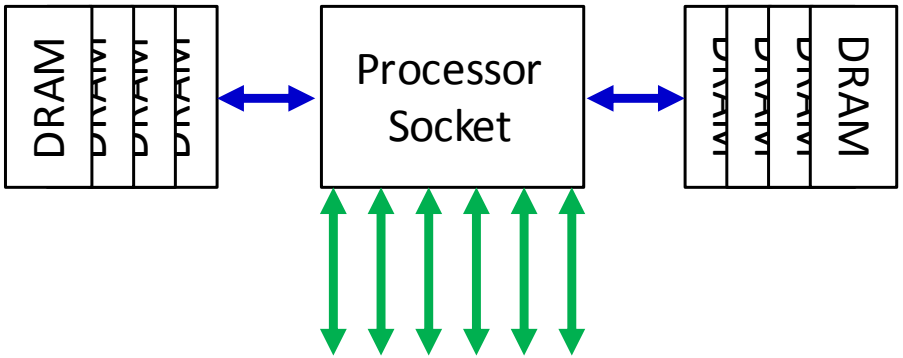


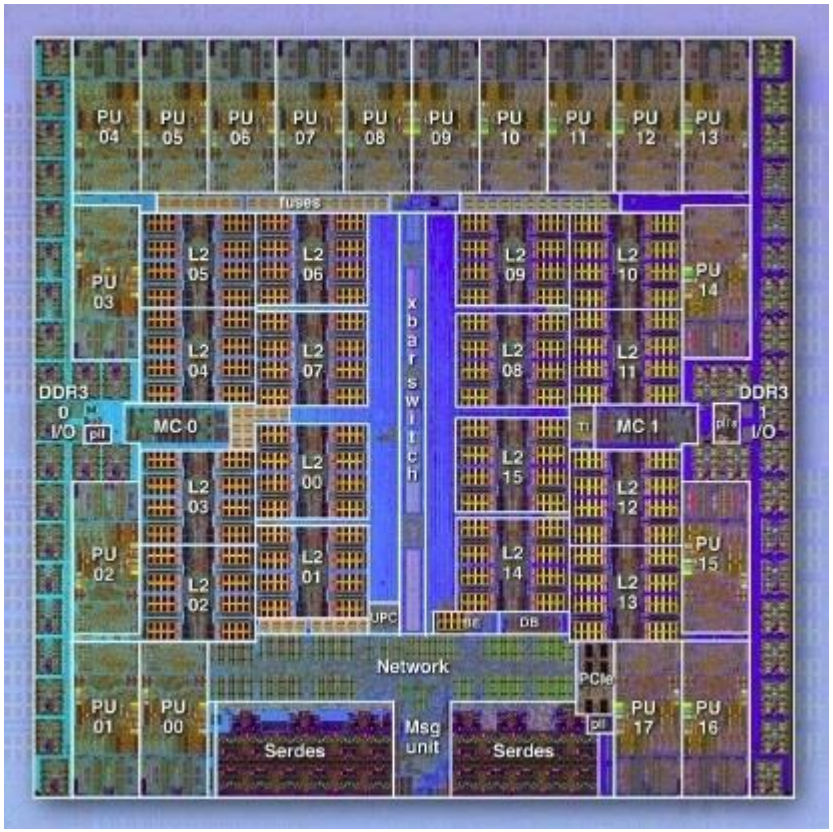
- **2 simple dual issue cores**
- **Each with dual FPUs**
- **Memory controller**
- **Large eDRAM L3**
- **3D message interface**
- **Collective interface**
- **All at subGHz clock**

"Packaging the Blue Gene/L supercomputer," IBM J. R&D, March/May 2005
"Blue Gene/L compute chip: Synthesis, timing, and physical design," IBM J. R&D, March/May 2005

BlueGene/Q





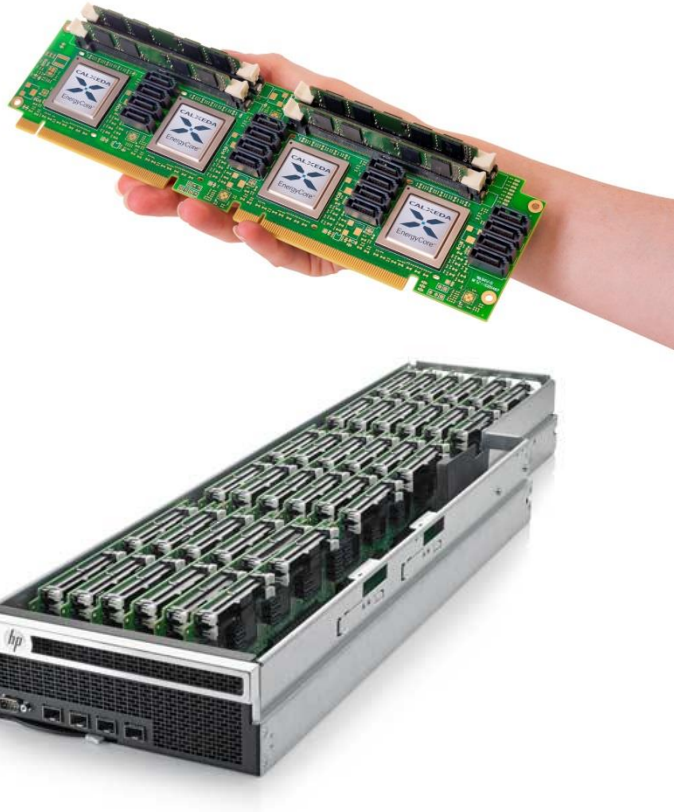


Integrated

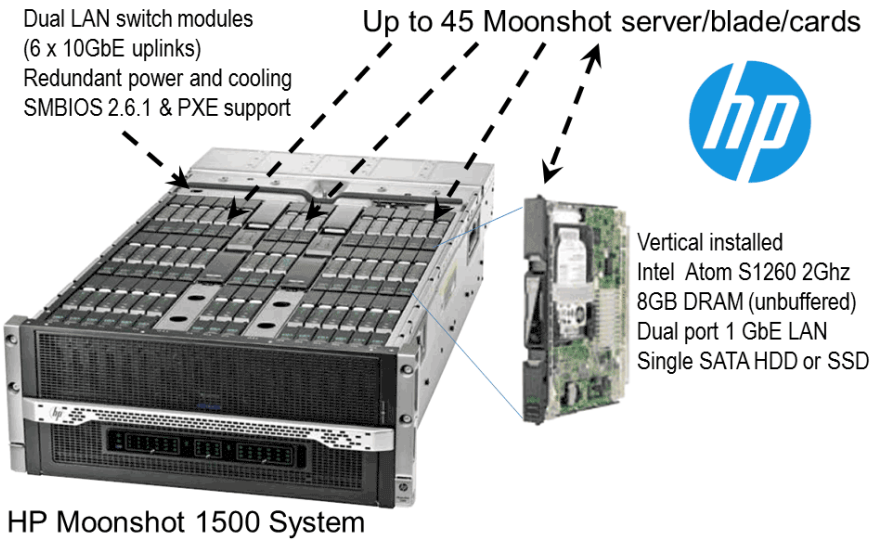
- **NIC**
- **Memory controllers**

http://www.heise.de/newsticker/meldung/SC-2010-IBM-zeigt-BlueGene-Q-mit-17-Kernen-1138226.html

Other Lightweight Systems Emerging



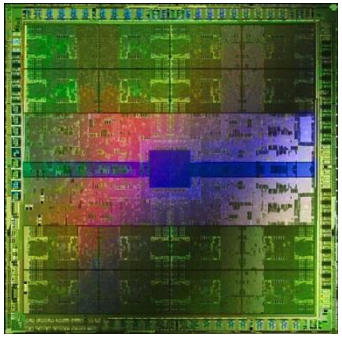
Calxeda quad-socket, quad-core ARMs



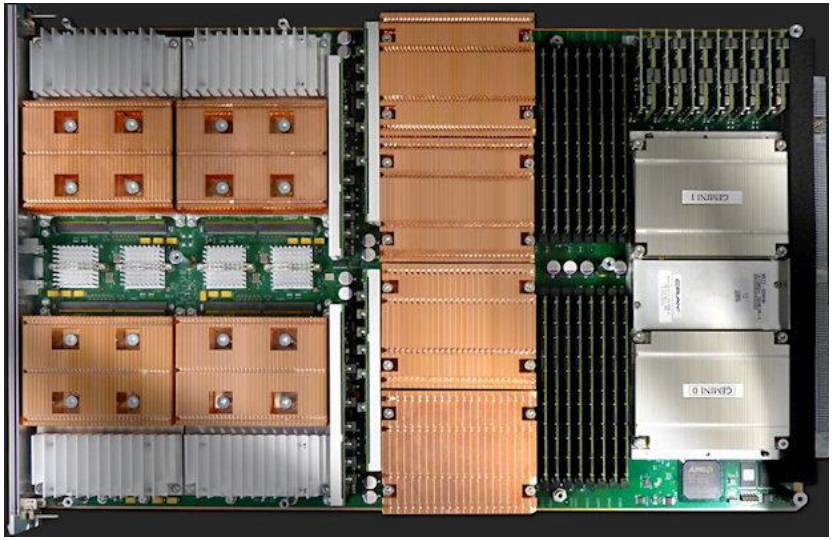
HP Moonshot 1500 System

HP Moonshot

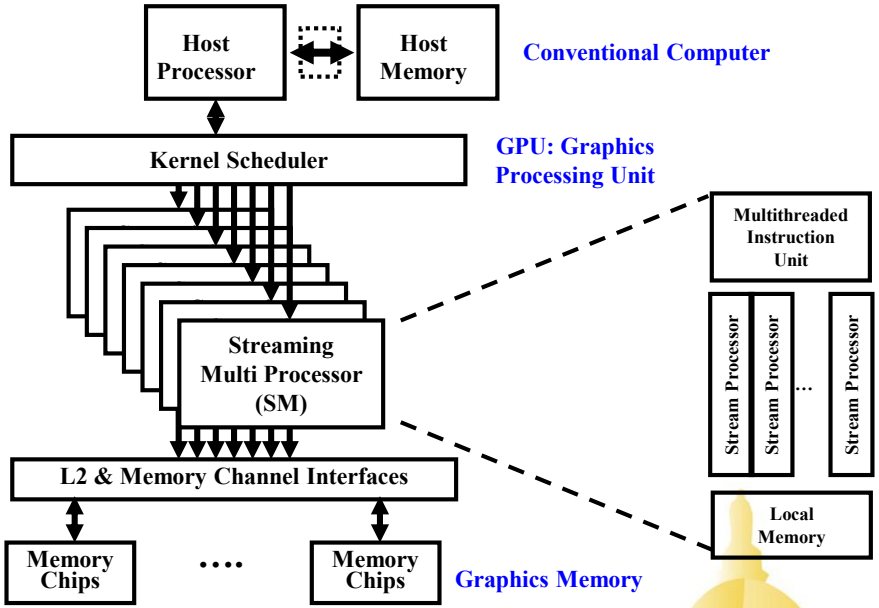
Heterogeneous Architectures



http://www.nvidia.com/object/fermi_architecture.html



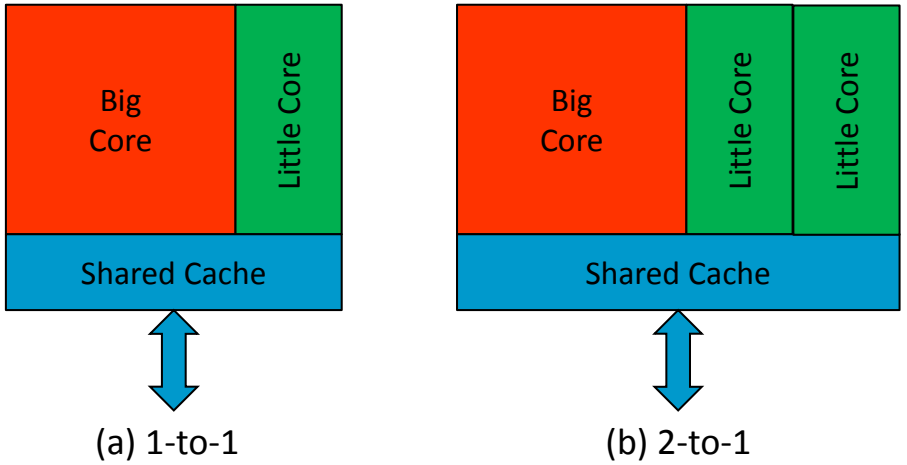
A Titan Blade



http://www.nvidia.com/docs/IO/43395/BD-05238-001_v03.pdf

Big Little Architectures

- Heterogeneous multi-core with same ISA
- "Bigger" cores:
  - Higher performance
  - But less energy efficient
- "Littler" cores:
  - Less performance
  - But more energy efficient

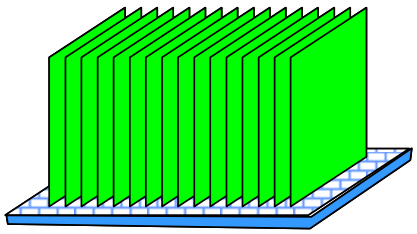


- Ability to move program states from core to core
- Examples:
  - ARM Cortex-A15 and A7, A53 and A57
  - *Almost* Intel Xeon and Xeon PHI

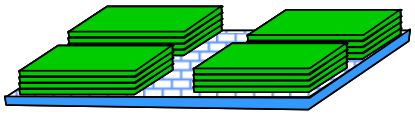
Topics

- 2004: Technology "Perfect Storm"
- Our Current Architecture Spectrum
- **The Exascale Memory Energy Horror Show**
- TOP500 Lessons
- GRAPH500 Lessons
- HPCC Lessons
- 3D Stack Projections

The Exascale Study Analysis: 67MW for 1EF/s = 67pj/flop

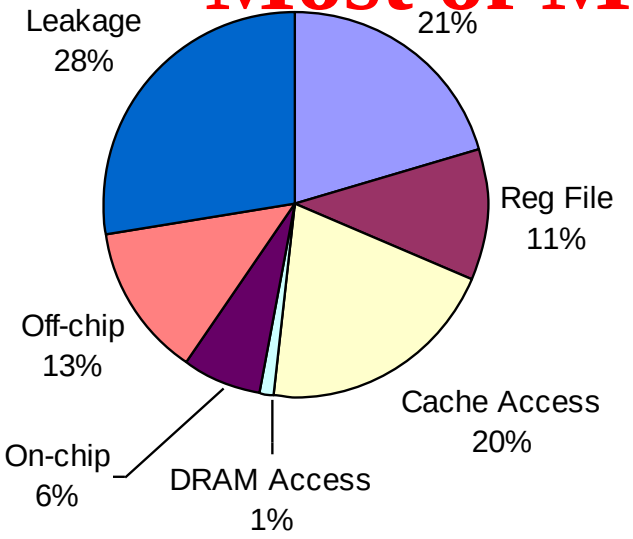


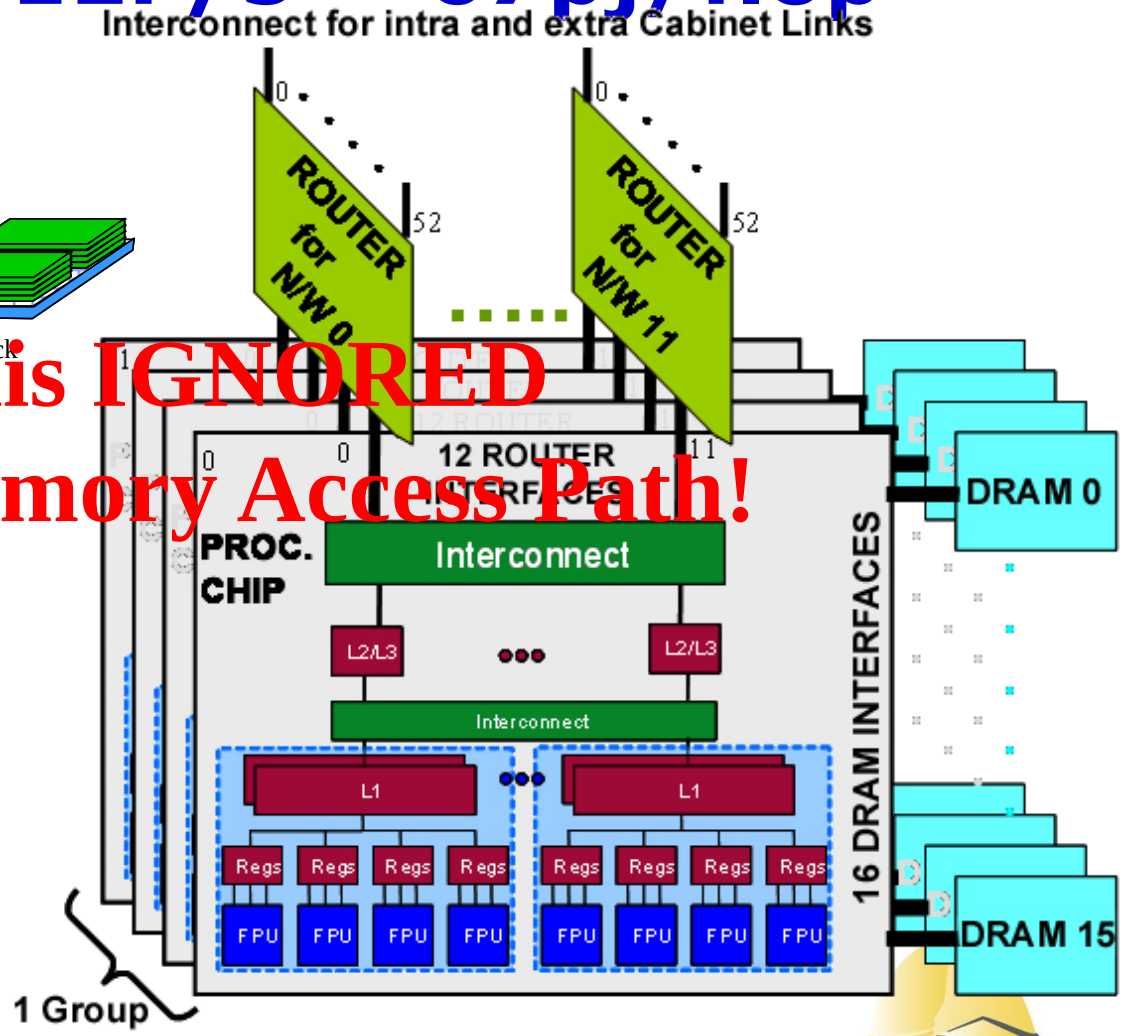
(a) Quilt Packaging



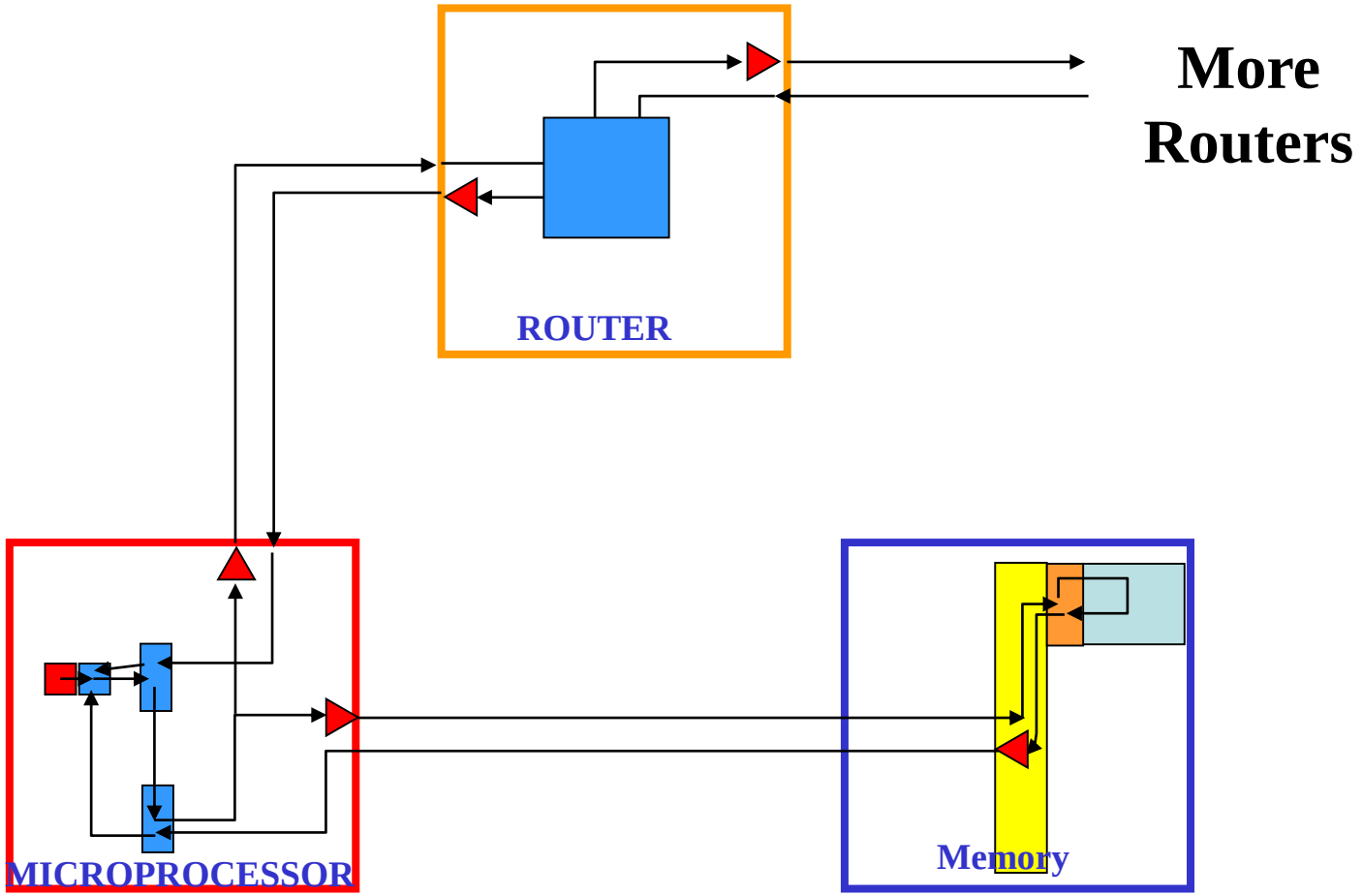
(b) Thru via chip stack

But This IGNORED Most of Memory Access Path!





The Access Path: Interconnect-Driven



Some sort of memory structure

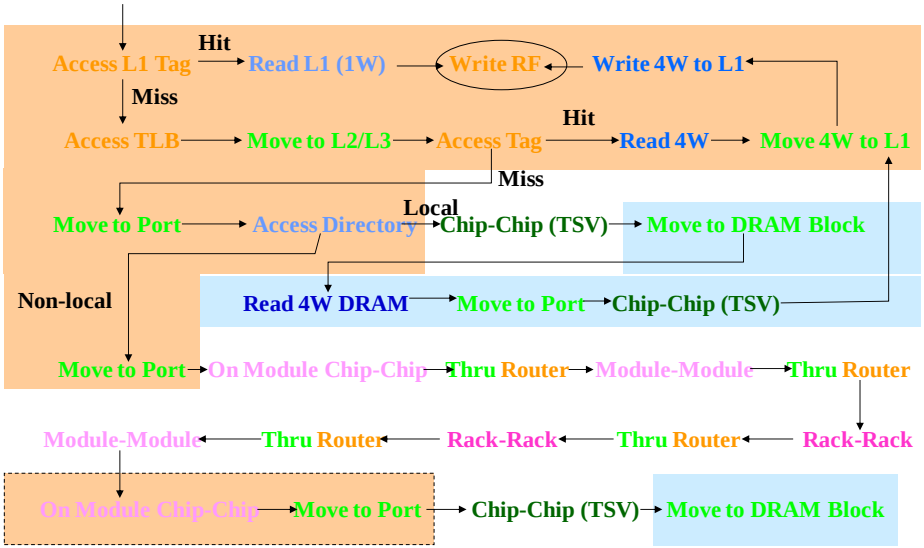
Sample Path – Off Module Access

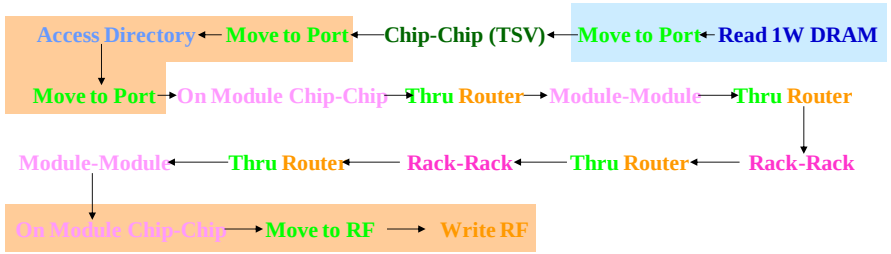
1. Check local L1 (miss)
2. Go thru TLB to remote L3 (miss)
3. Across chip to correct port (thru routing table RAM)
4. Off-chip to router chip
5. 3 times thru router and out
6. Across microprocessor chip to correct DRAM I/F
7. Off-chip to get to correct DRAM chip
8. Cross DRAM chip to correct array block
9. Access DRAM Array
10. Return data to correct I/R
11. Off-chip to return data to microprocessor
12. Across chip to Routre Table
13. Across microprocessor to correct I/O port
14. Off-chip to correct router chip
15. 3 times thru router and out
16. Across microprocessor to correct core
17. Save in L2, L1 as required
18. Into Register File

Taper Data from Exascale Report

Memory Level	Capacity GB	Bandwidth GB/s	BW Taper Operands/clock	Power mW/flop	Energy Taper Operands/pJ
Register File	1KB	35,960	4	8.2	0.56
L1	64KB	8,992	0.25	5.5	0.068
L2	256KB	4,496	0.125	3.5	0.054
L3	97MB	2,288	0.0625	3.8	0.025
Local DRAM	16GB	712	0.02	10.5	0.0029
Network DRAM	3.8PB	216	0.006	11.5	0.00008

Relook at Exascale Strawman





AND THIS DOESN'T ACCOUNT FOR TLB MISSES!!!

In 2015, core energy per flop for Linpack is < 10pJ

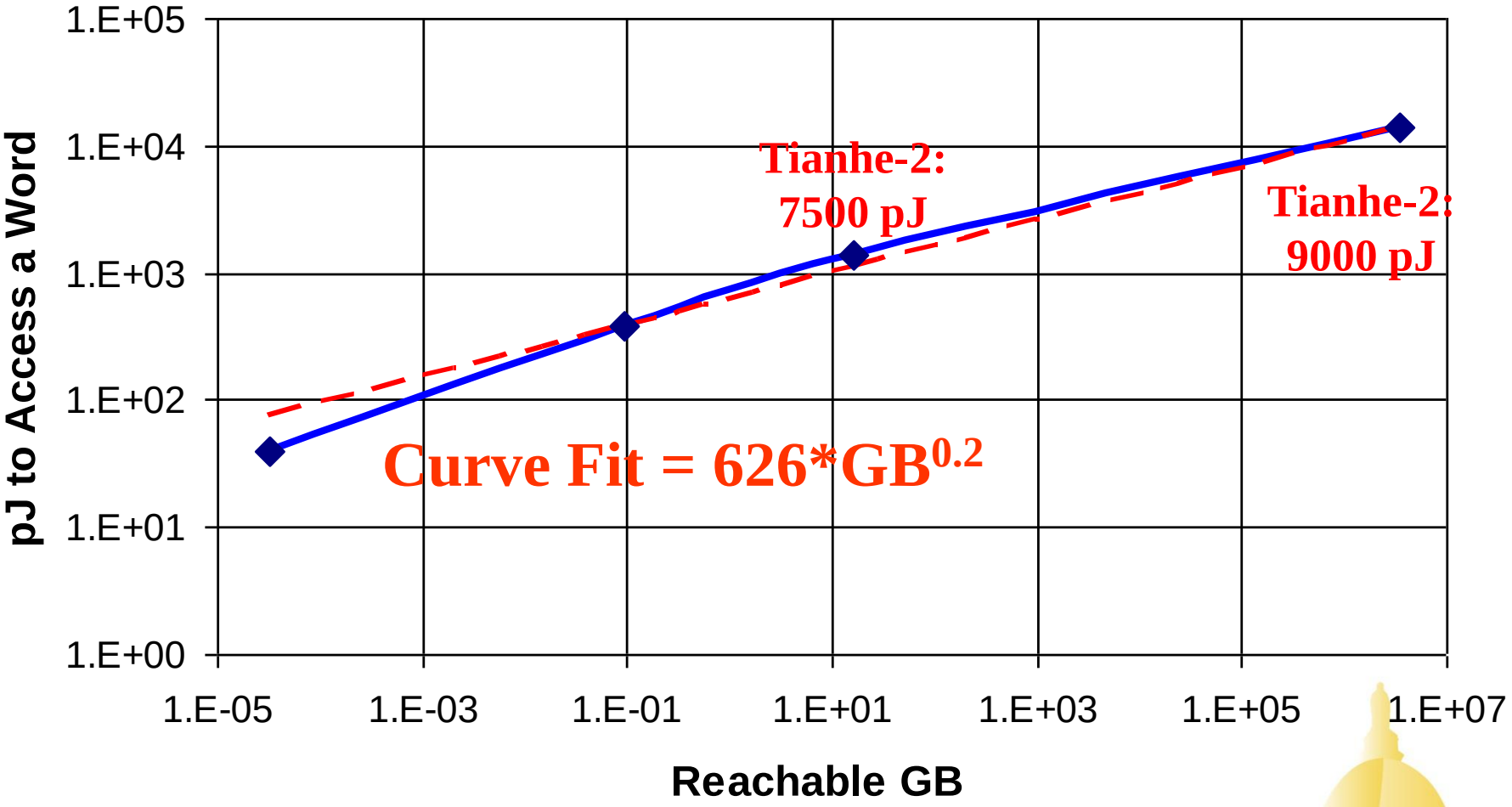
Operation	Energy (pJ/bit)
Register File Access	0.16
SRAM Access	0.23
DRAM Access	1
On-chip movement	0.0187
Thru Silicon Vias (TSV)	0.011
Chip-to-Board	2
Chip-to-optical	10
Router on-chip	2

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	121,440	36%
Read Other Y[i]s	L1	39	264	10,296	3%
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	

50X

If this is true, 1 EF/s = 0.5 GW!

Access vs Reach



And This Means You Can Reference Memory Occasionally….

- 100% L1 Hit, once every 6.5 Flops
- 80% L1 Hit and 100% L2/L3 Hit, once every 18 Flops
- 80% L1 Hit, 90% L2/L3 Hit, and all the rest is local, once every 35 Flops
- 80% L1 Hit, 90% L2/L3 Hit, 60% local, and 40% Global, once every 118 Flops

What Does This Tell Us?

- Cannot afford ***ANY*** memory references
- Many more energy sinks than you think
- Cost of Interconnect ***Dominates***
- Must design for on-board or stacked DRAM
- Need to redesign the entire access path:
  - Alternative memory technologies – reduce access cost
  - Alternative packaging costs – reduce bit movement cost
  - Alternative transport protocols – reduce # bits moved
  - Alternative execution models – reduce # of movements

AND IT GETS *MUCH WORSE*
FOR CACHE UNFRIENDLY PROBLEMS

The Key Take Away

You can hide the latency…

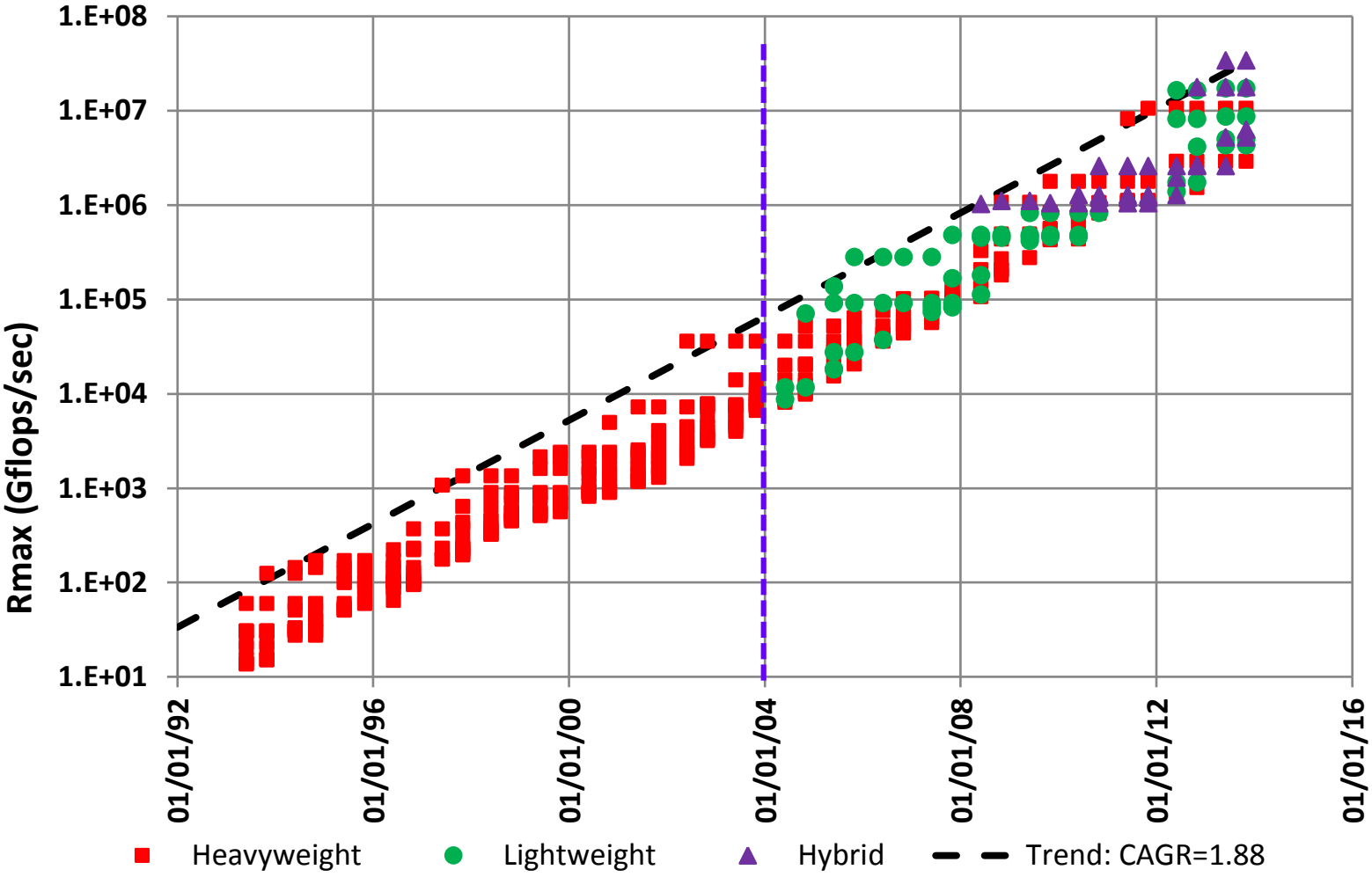
But

You <u>Can't Hide</u> the Energy

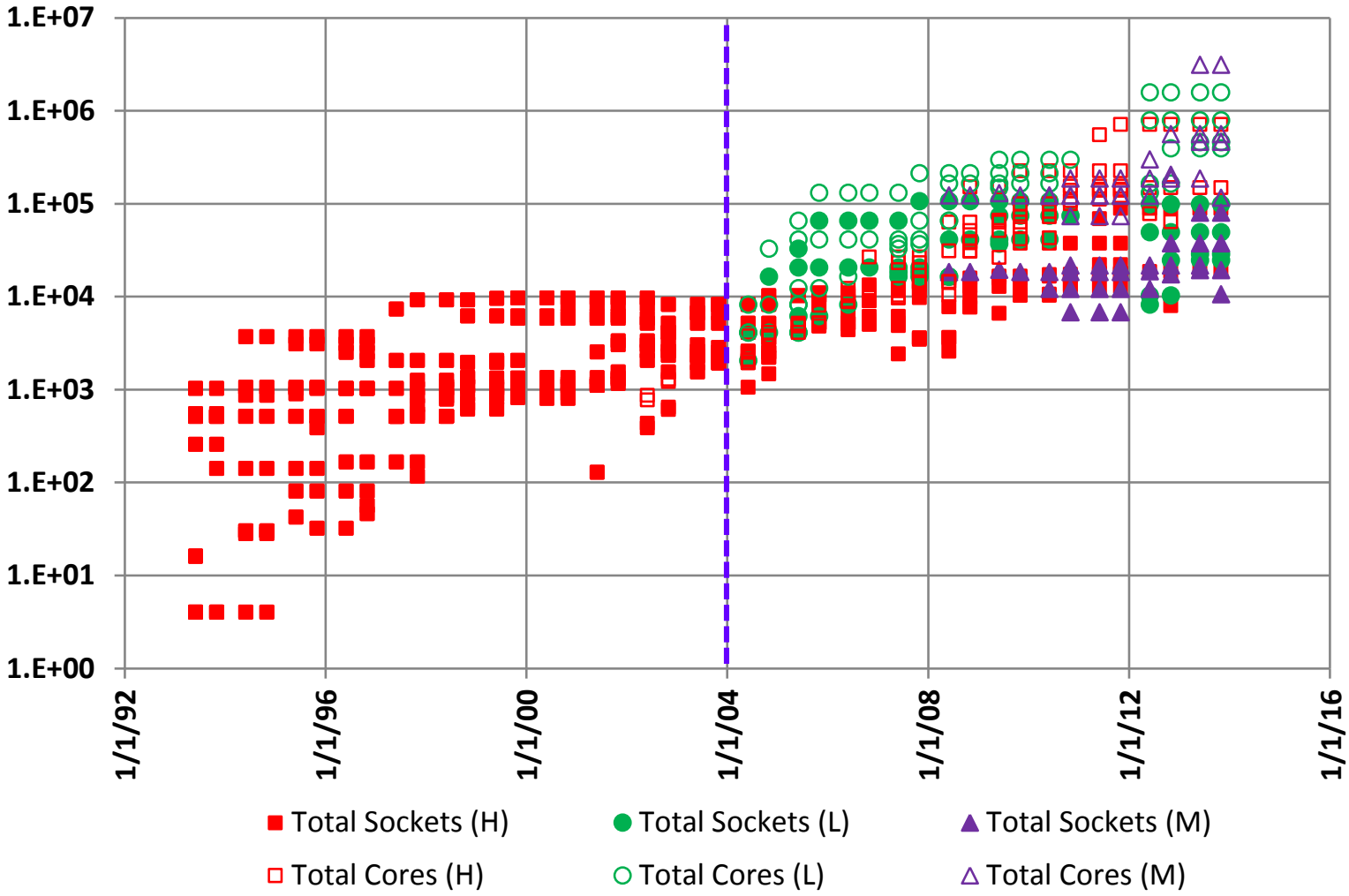
Topics

- 2004: Technology "Perfect Storm"
- Our Current Architecture Spectrum
- The Exascale Memory Energy Horror Show
- **TOP500 Lessons**
- GRAPH500 Lessons
- HPCC Lessons
- 3D Stack Projections

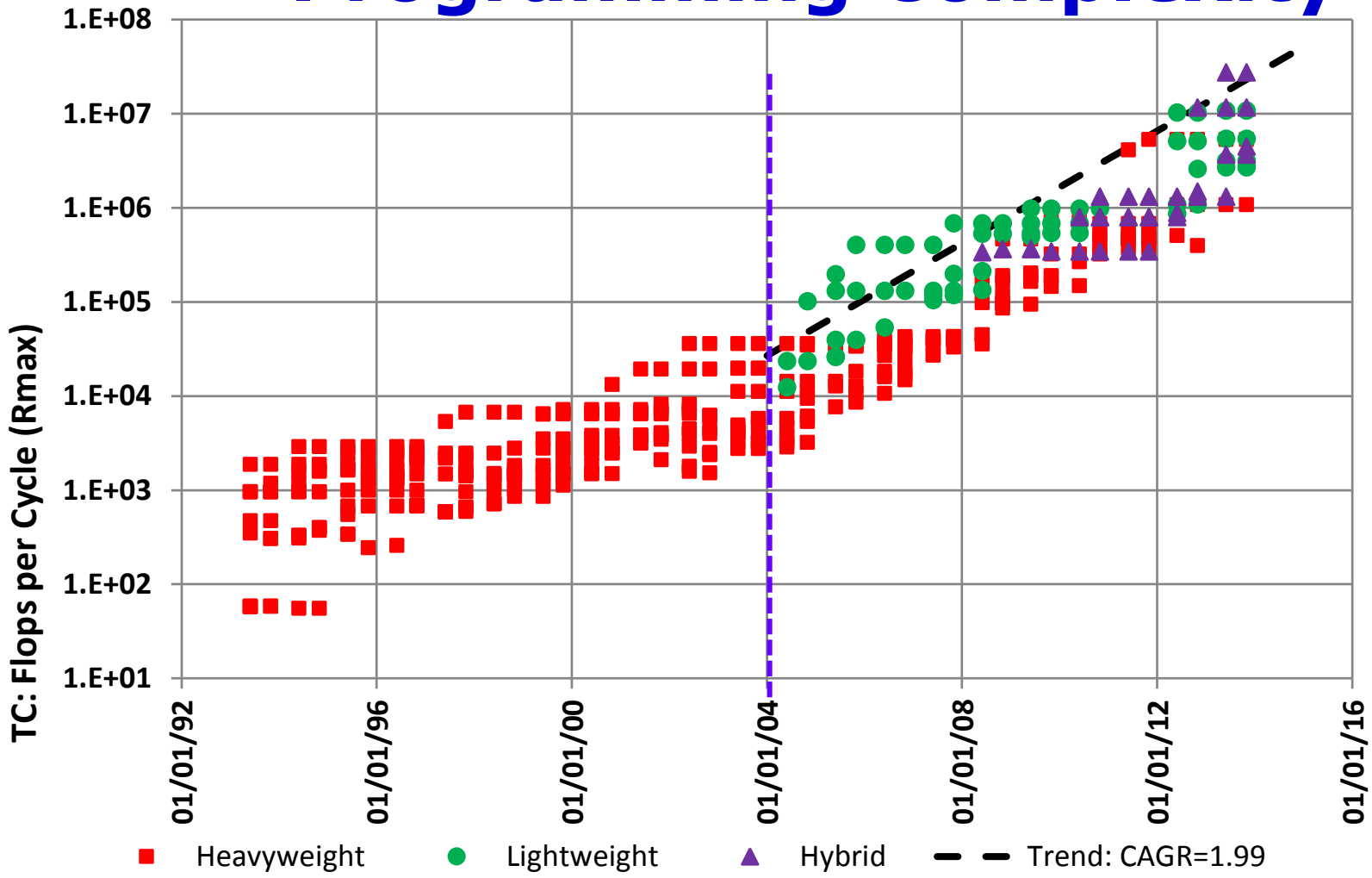
Emergence of Lightweight, Hybrid



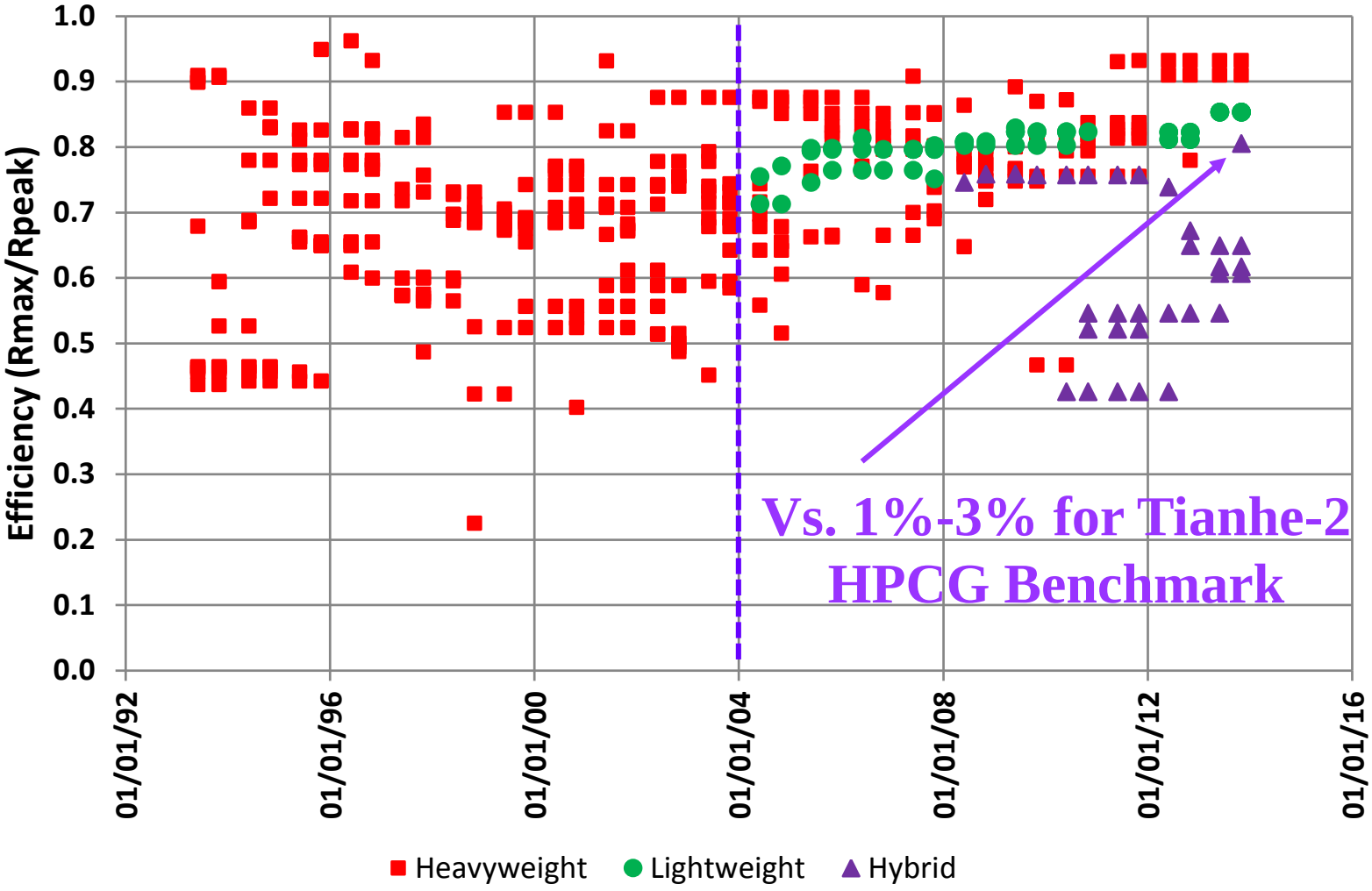
Sockets and Cores Growing



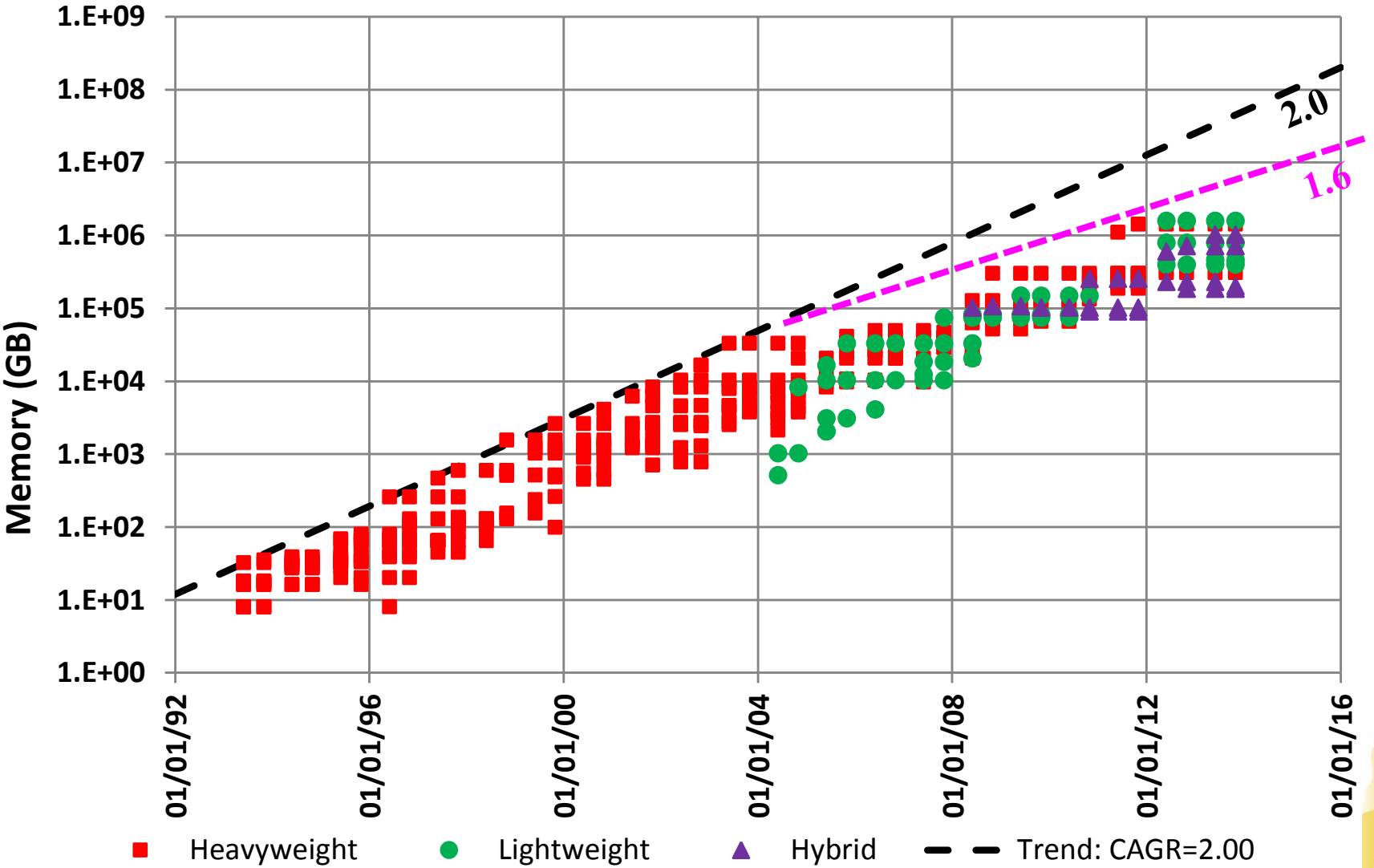
Flops/Cycle: This Drives Programming Complexity



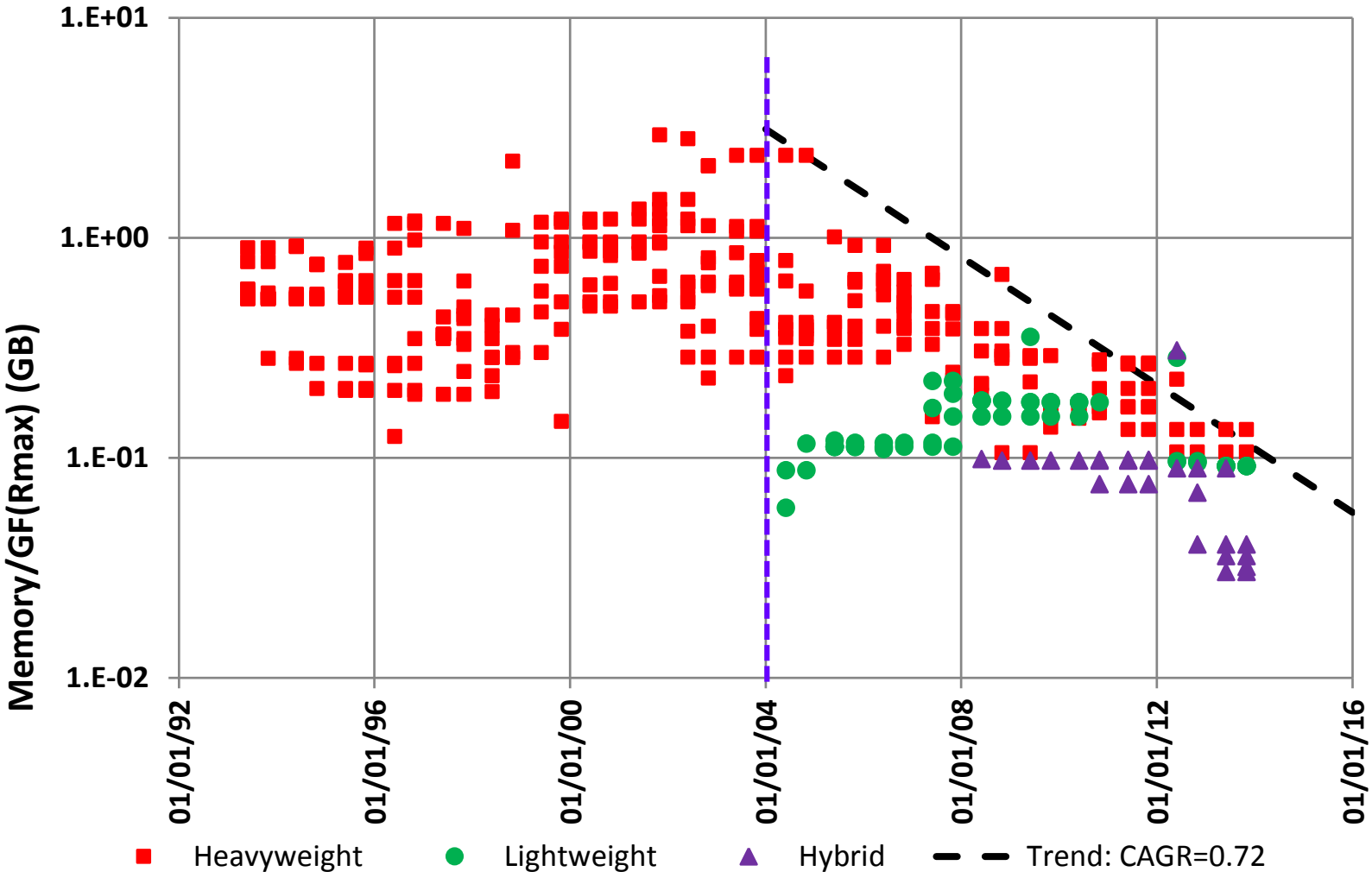
Floating Point Efficiency



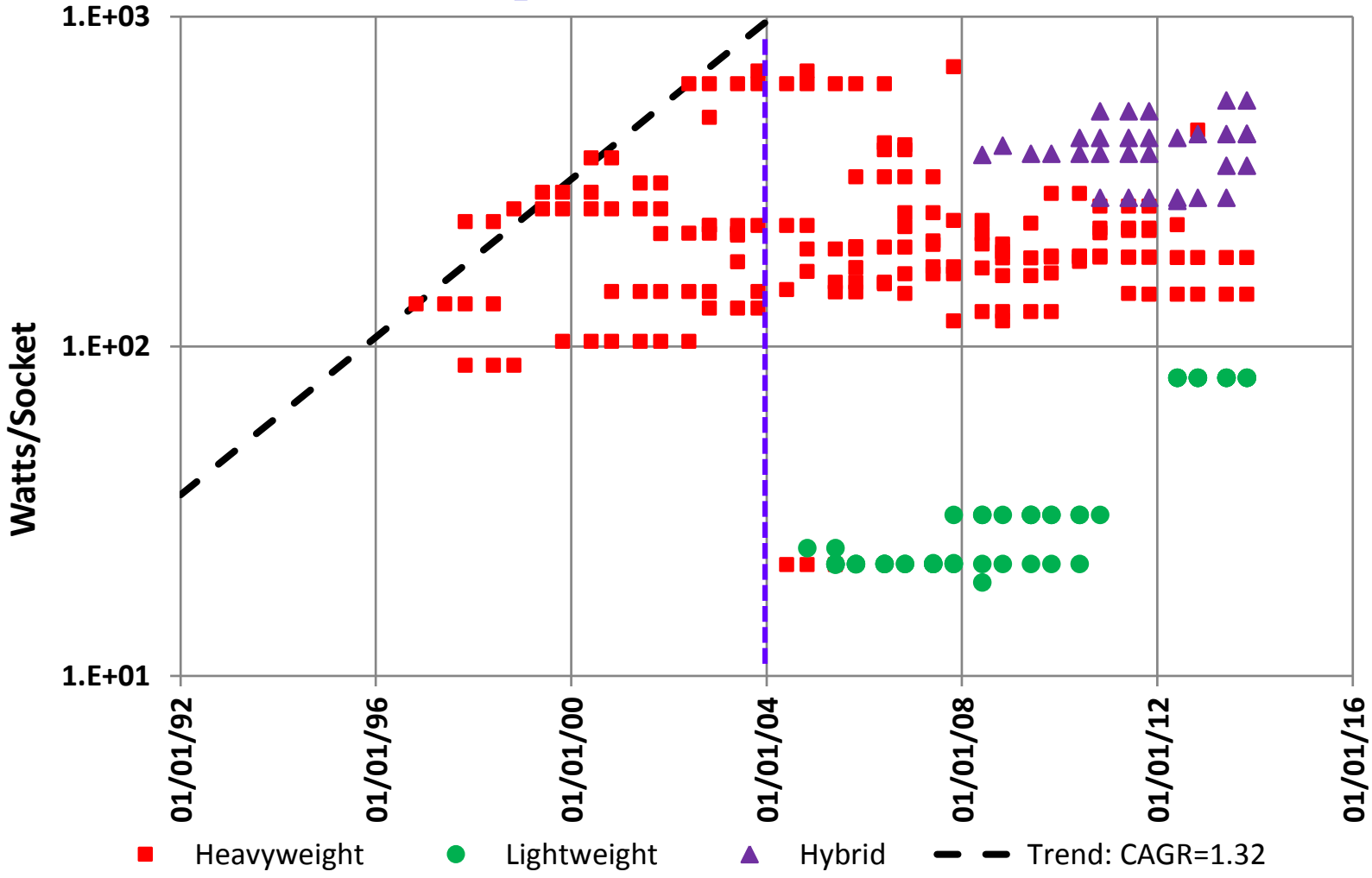
Memory Growth Has Slowed



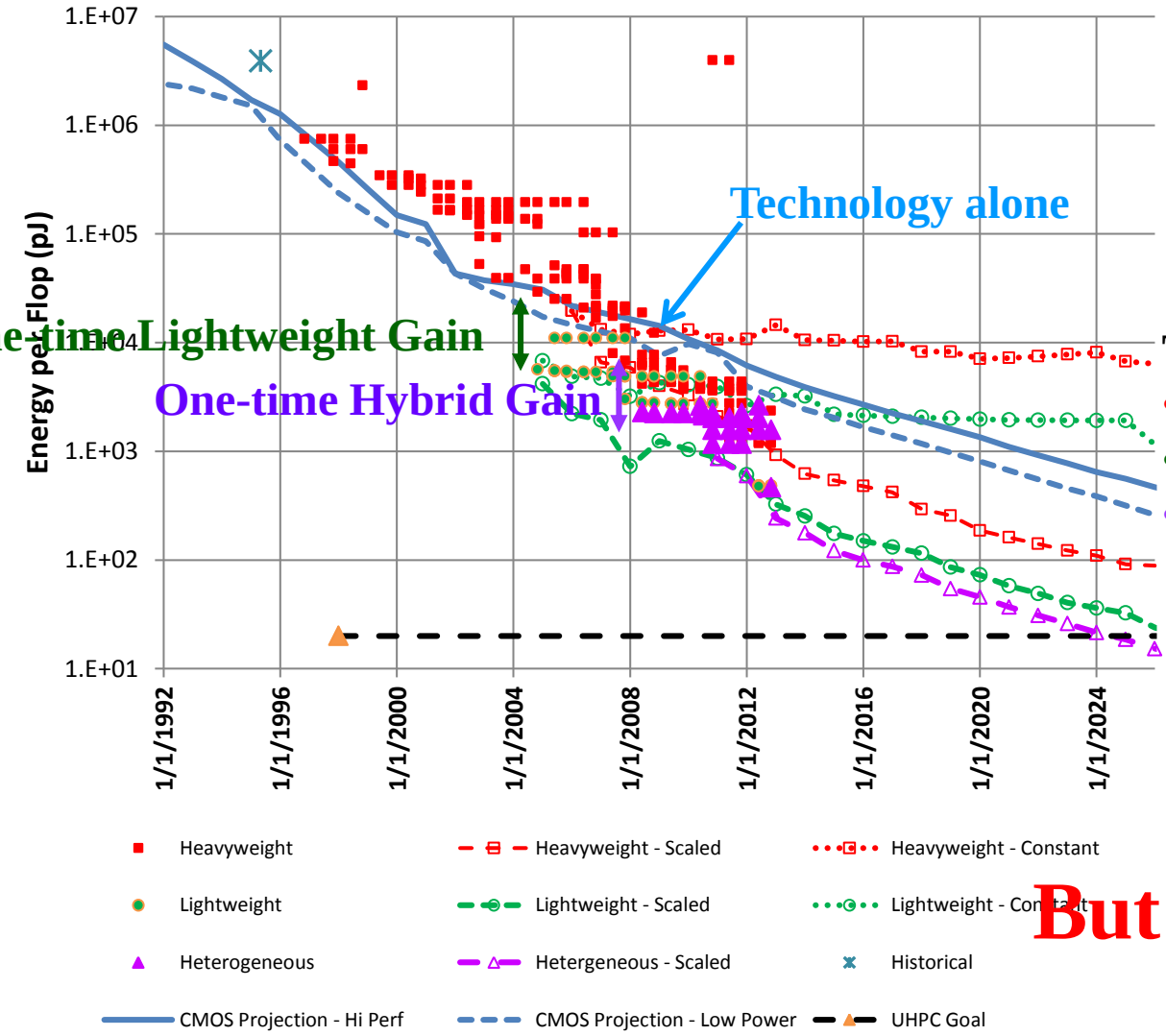
And Memory per Flop/s Is Dropping!



Ceiling on System Power "per Socket"



Energy per Flop is Dropping

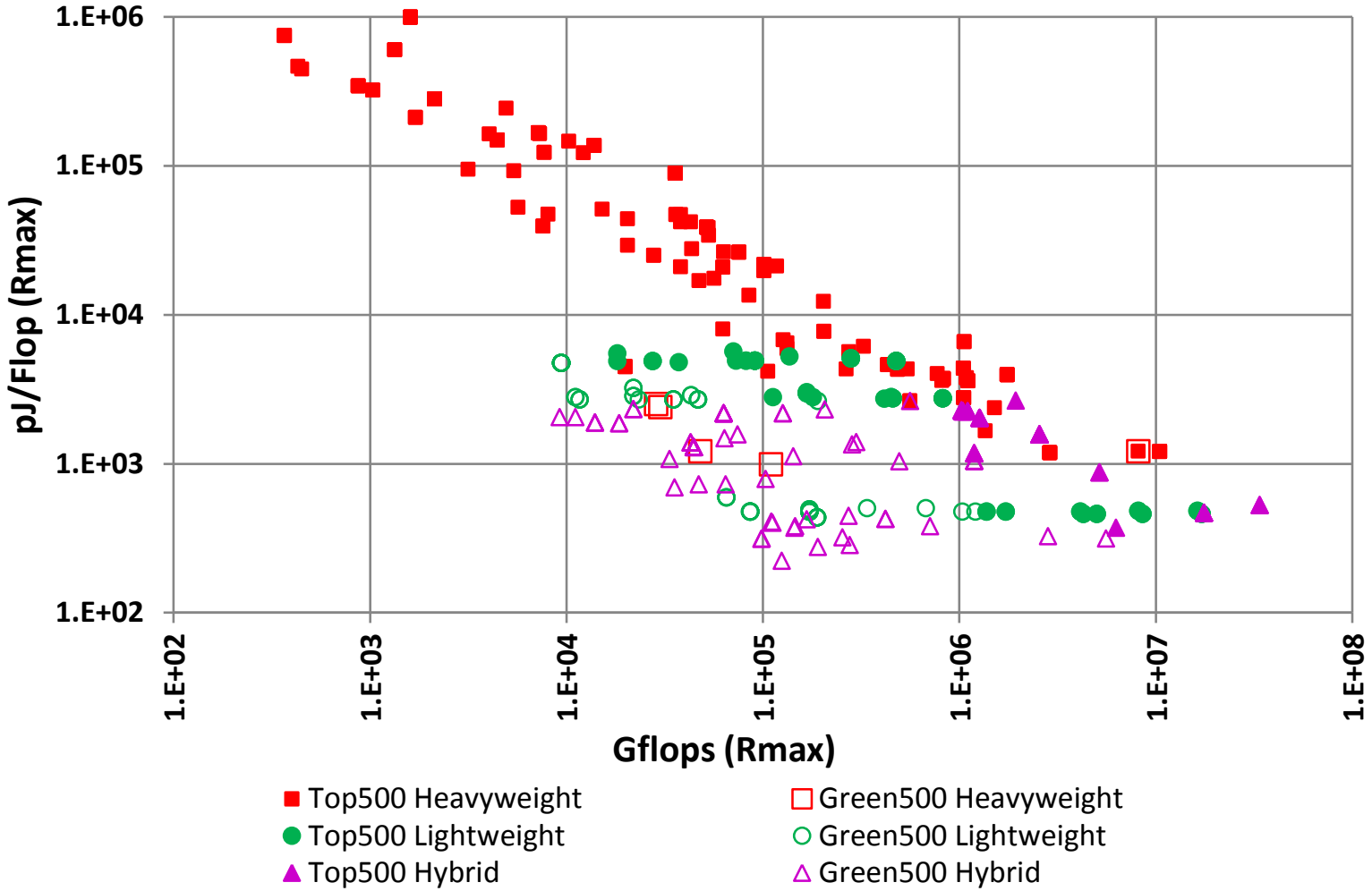


Trends vs 20pJ/Flop goal

- **Heavyweight: never**
- **Lightweight: close**
- **Hybrid: Possible**

But this is Rpeak

But Energy Efficiency Limits Performance



Conclusions

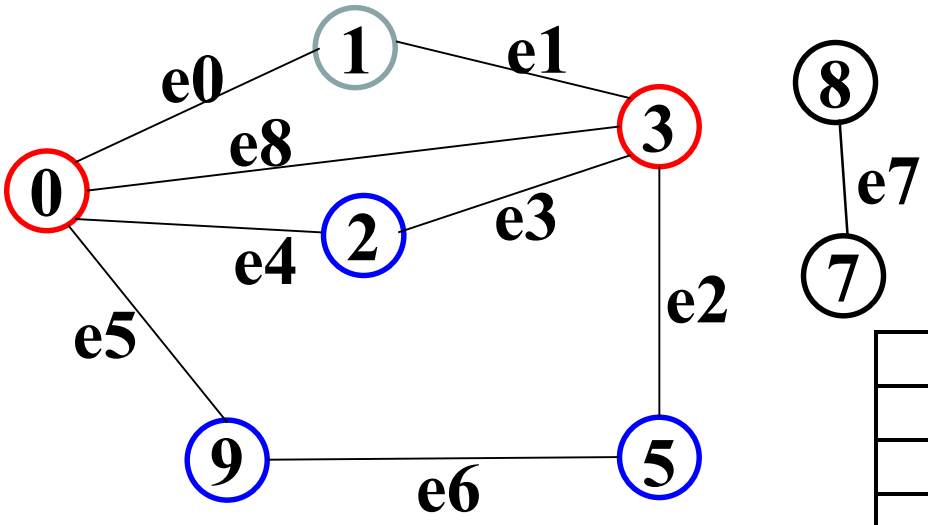
- TOP500 continues to grow at 2X/year
  - With 80% efficiency
- But less and less relevance to real problems
- And need to program for 10s of millions of flops/cycle
- With falling relative memory footprint
- New HPCG benchmark due out June

Topics

- 2004: Technology "Perfect Storm"
- Our Current Architecture Spectrum
- The Exascale Memory Energy Horror Show
- TOP500 Lessons
- **GRAPH500 Lessons**
- HPCC Lessons
- 3D Stack Projections

Graph 500

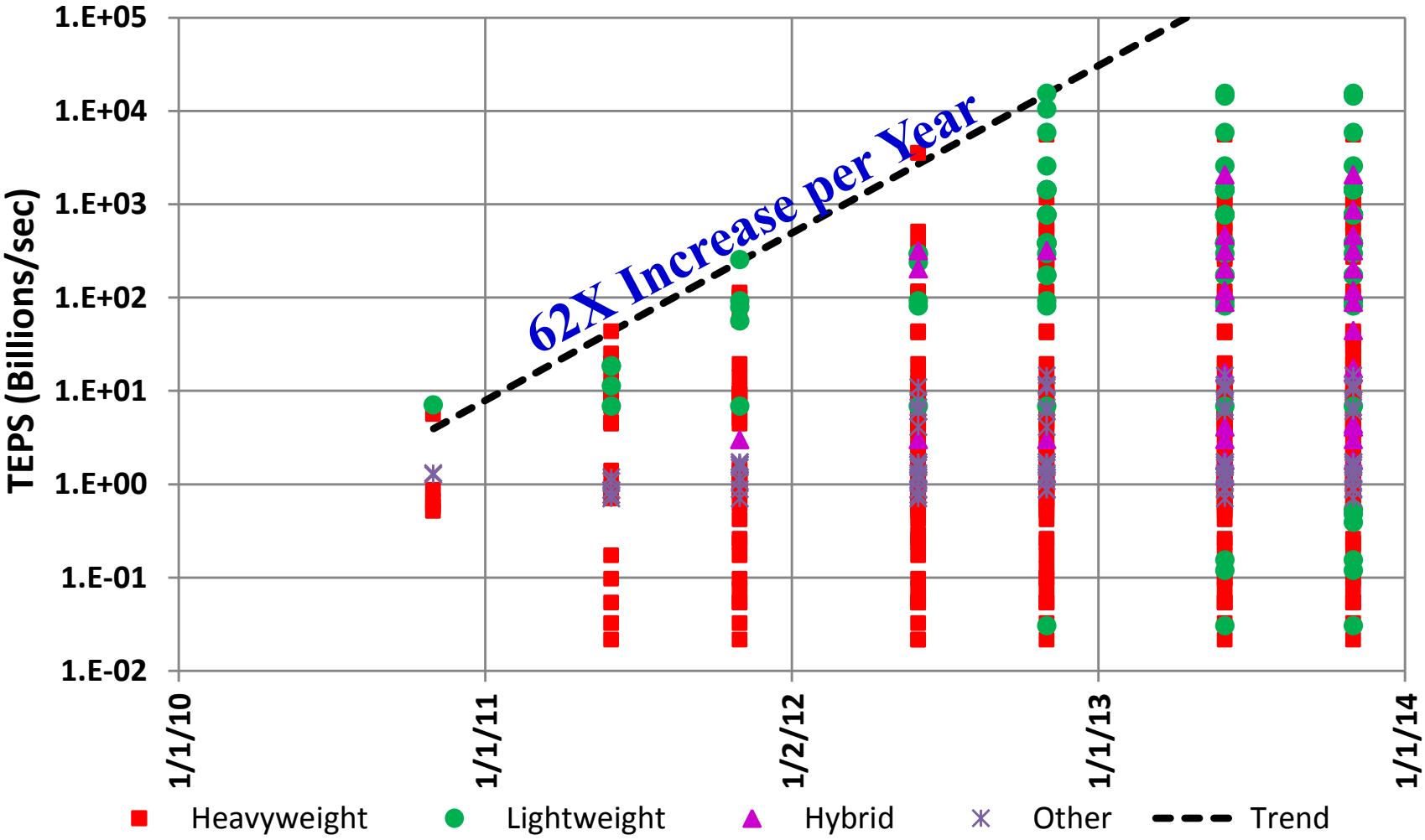
- Start with a root, find reachable nodes
- Simplifications: only 1 kind of edge, no weights
- Performance metric: **TEPS**: Traversed Edges/sec



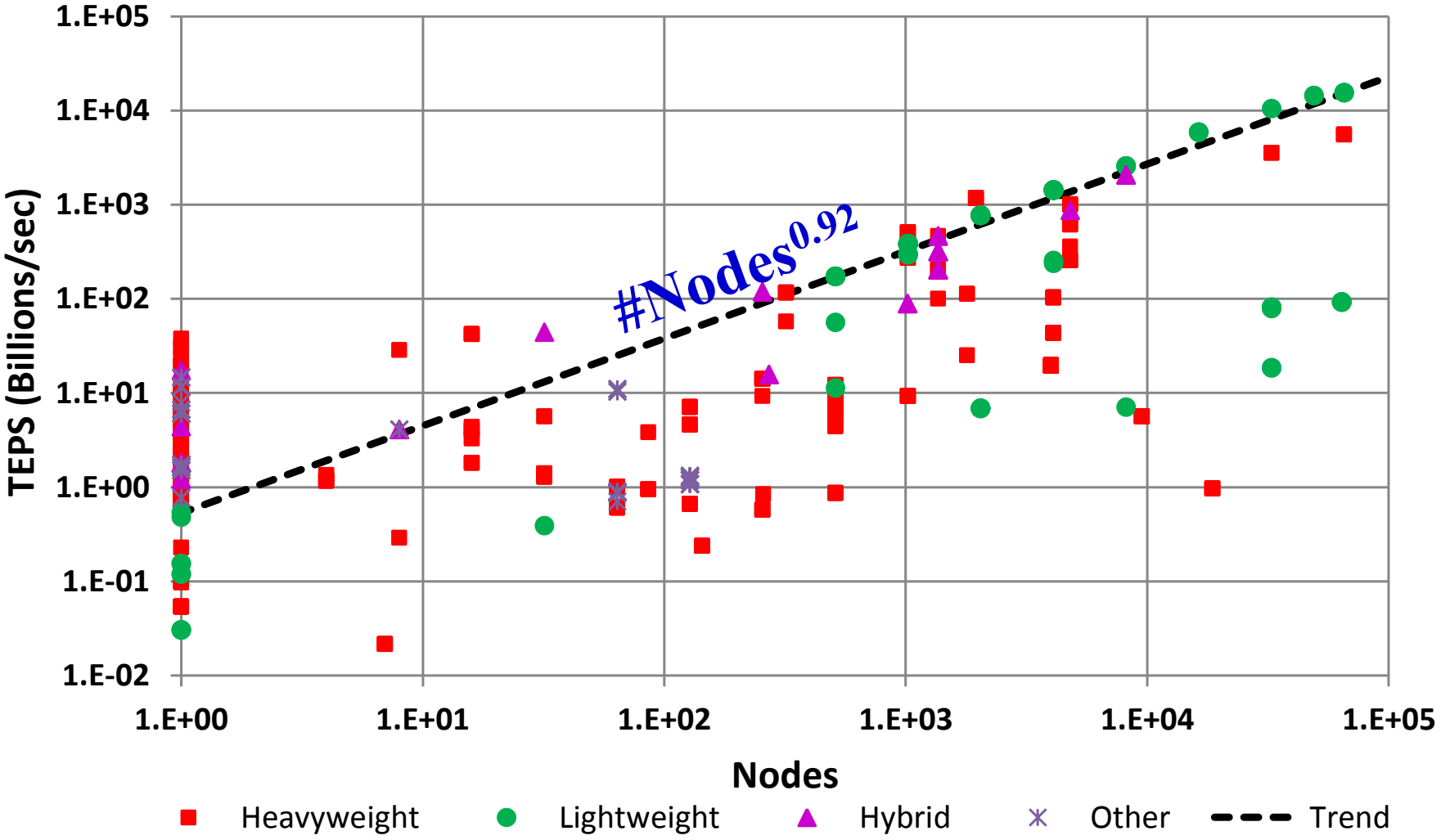
Starting at 1: 1, 0, 3, 2, 9, 5

Level	Scale	Size	GB	Edge(B)
10	26	Toy	17	272
11	29	Mini	140	280
12	32	Small	1024	256
13	36	Medium	17408	272
14	39	Large	143360	280
15	42	Huge	1153434	281.6
			Average	273.6

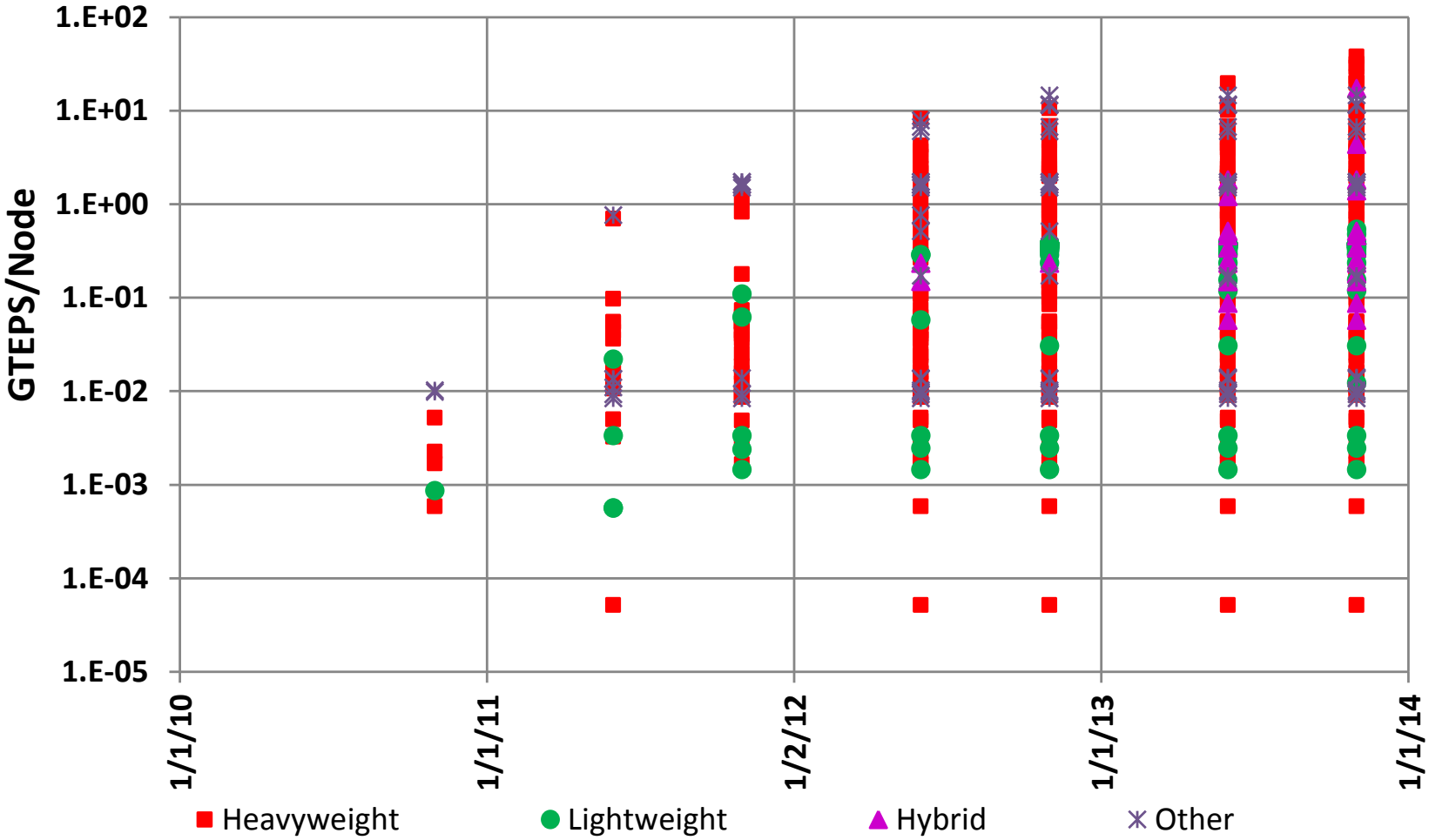
TEPS vs Time


1.E+05
1.E+04
1.E+03
1.E+02
1.E+01
1.E+00
1.E-01
1.E-02
TEPS (Billions/sec)
62X Increase per Year
1/1/10
1/1/11
1/2/12
1/1/13
1/1/14
Heavyweight
Lightweight
Hybrid
Other
Trend

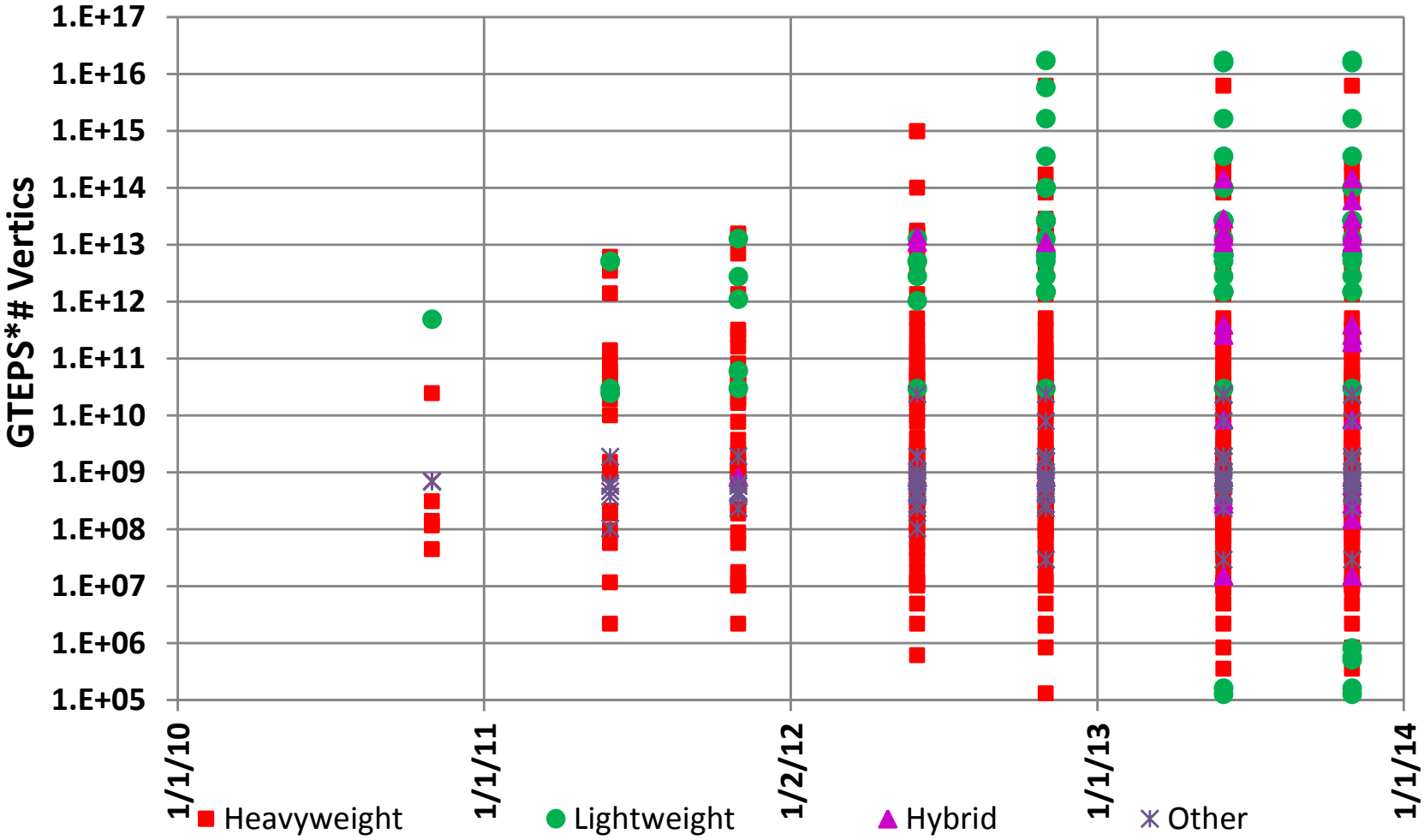
TEPS vs # Nodes



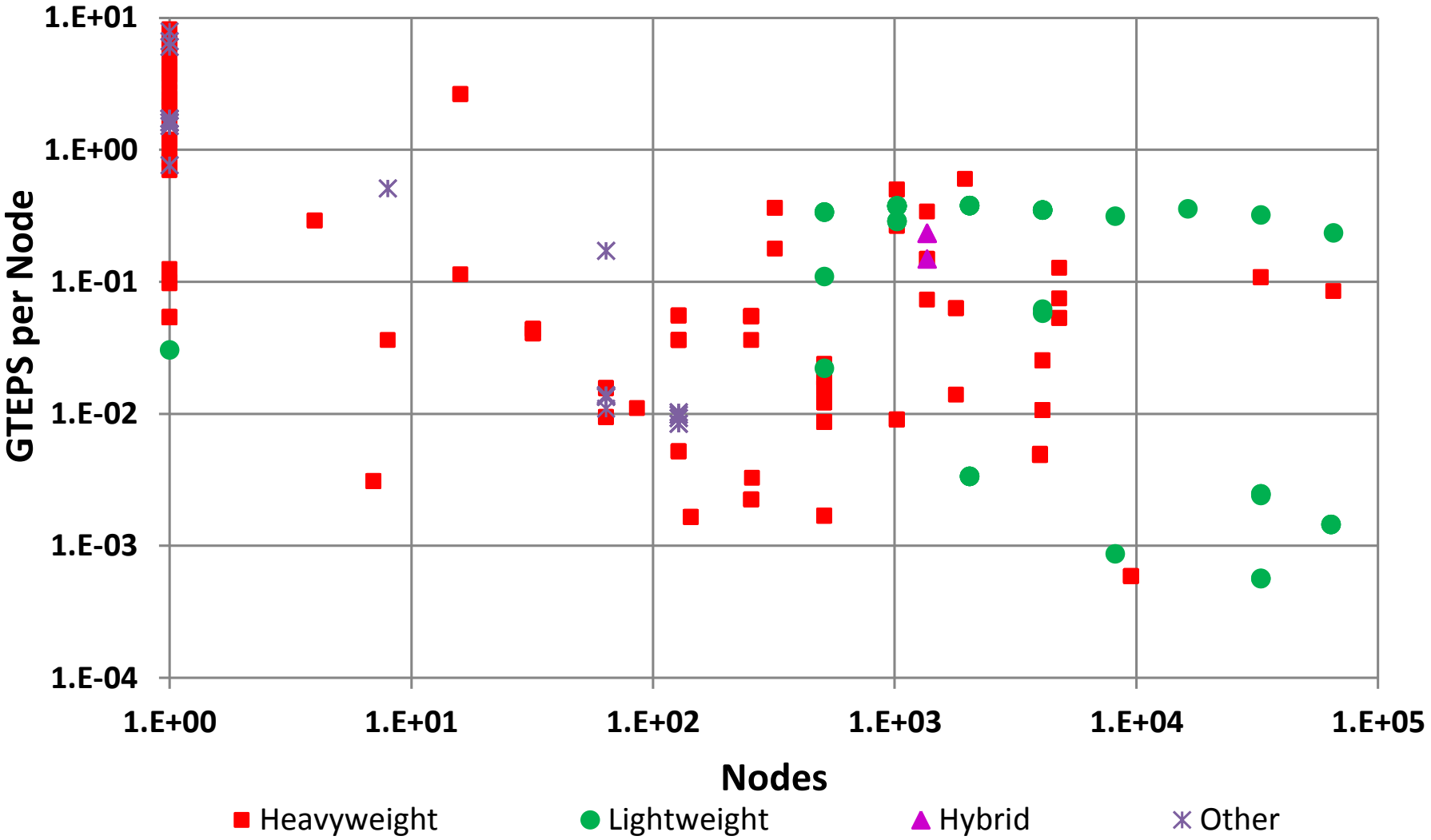
TEPS per Node vs Time



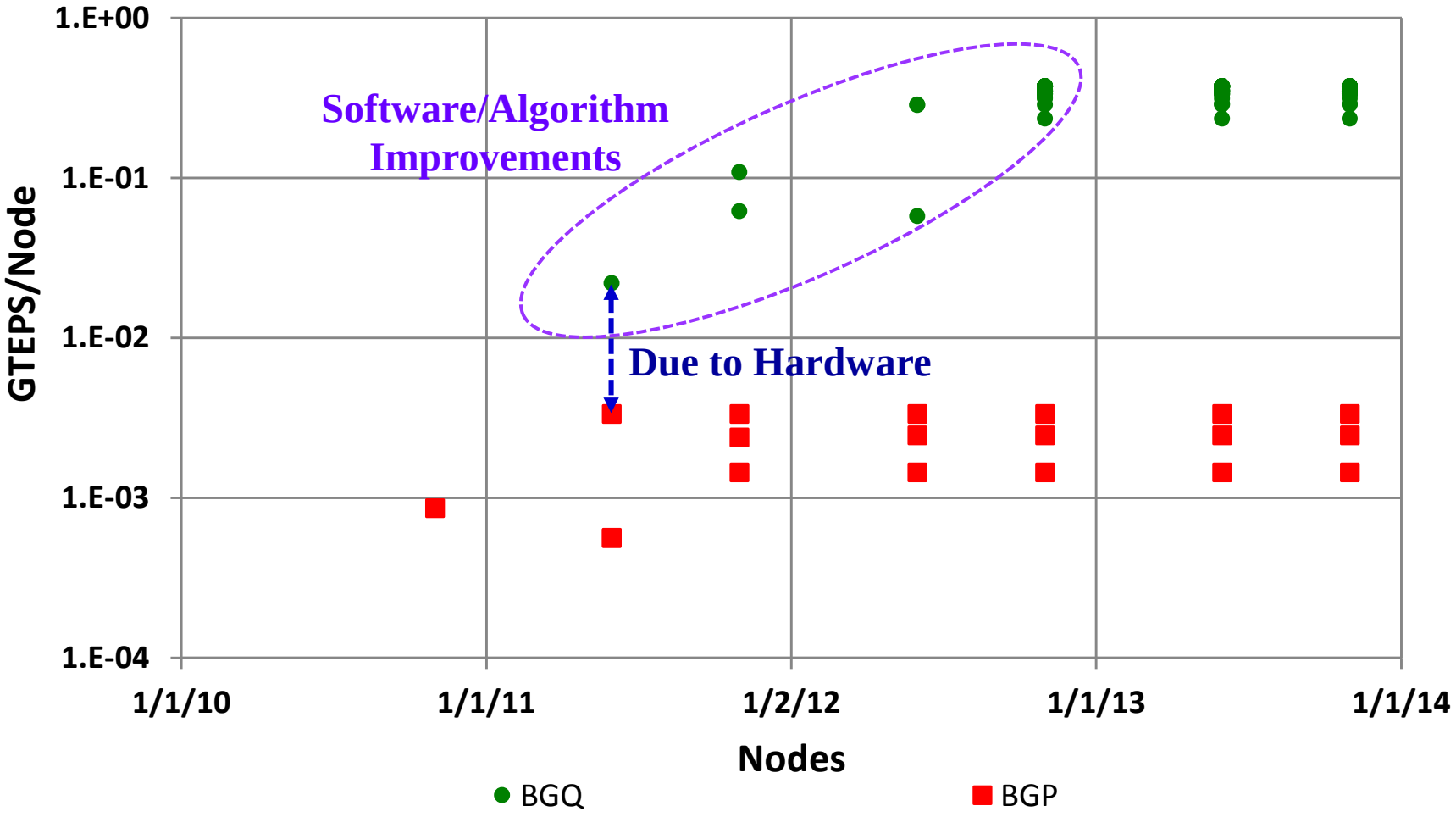
Combined Metric: TEPS*# Vertices



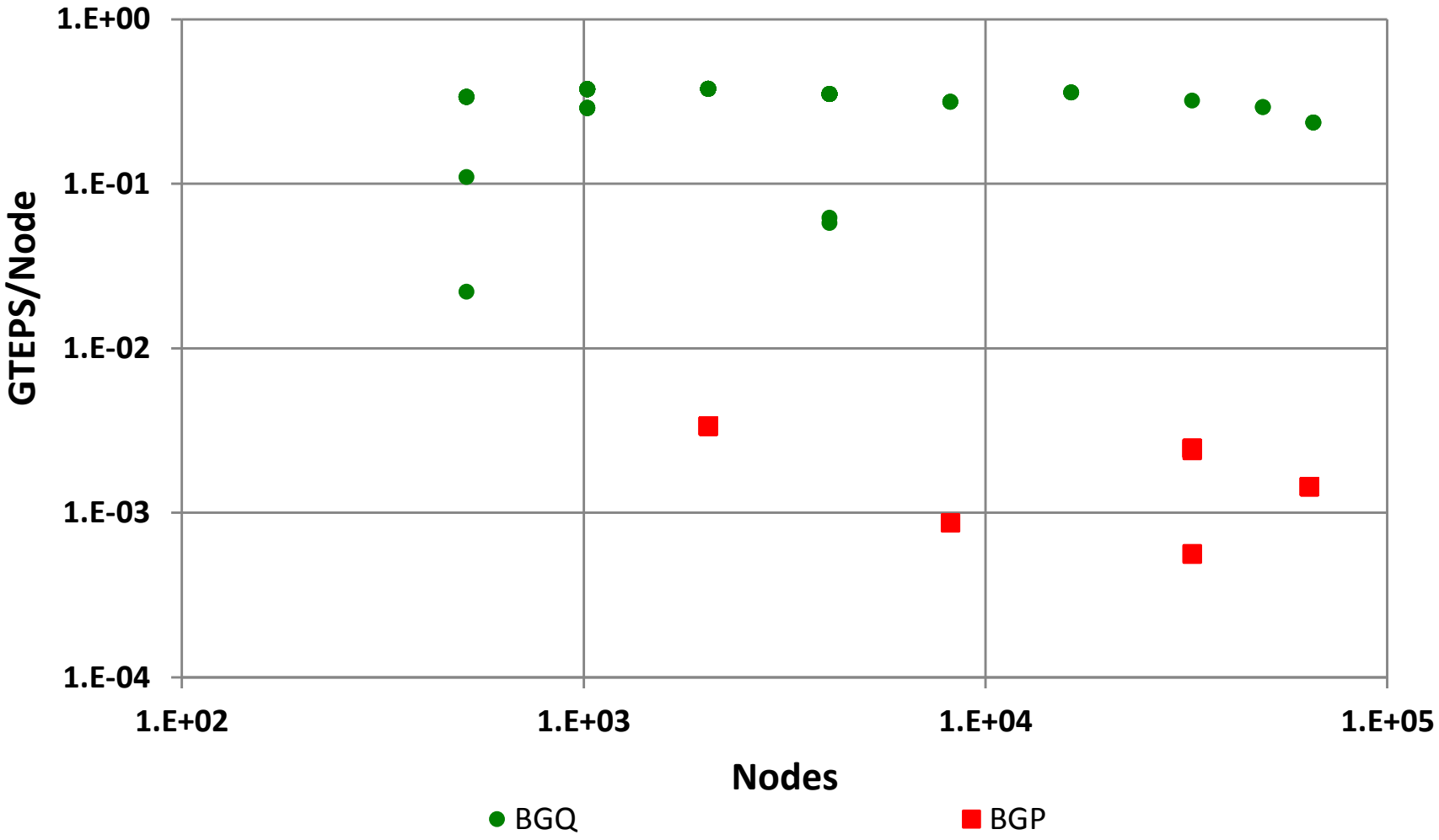
TEPS per Node vs # Nodes



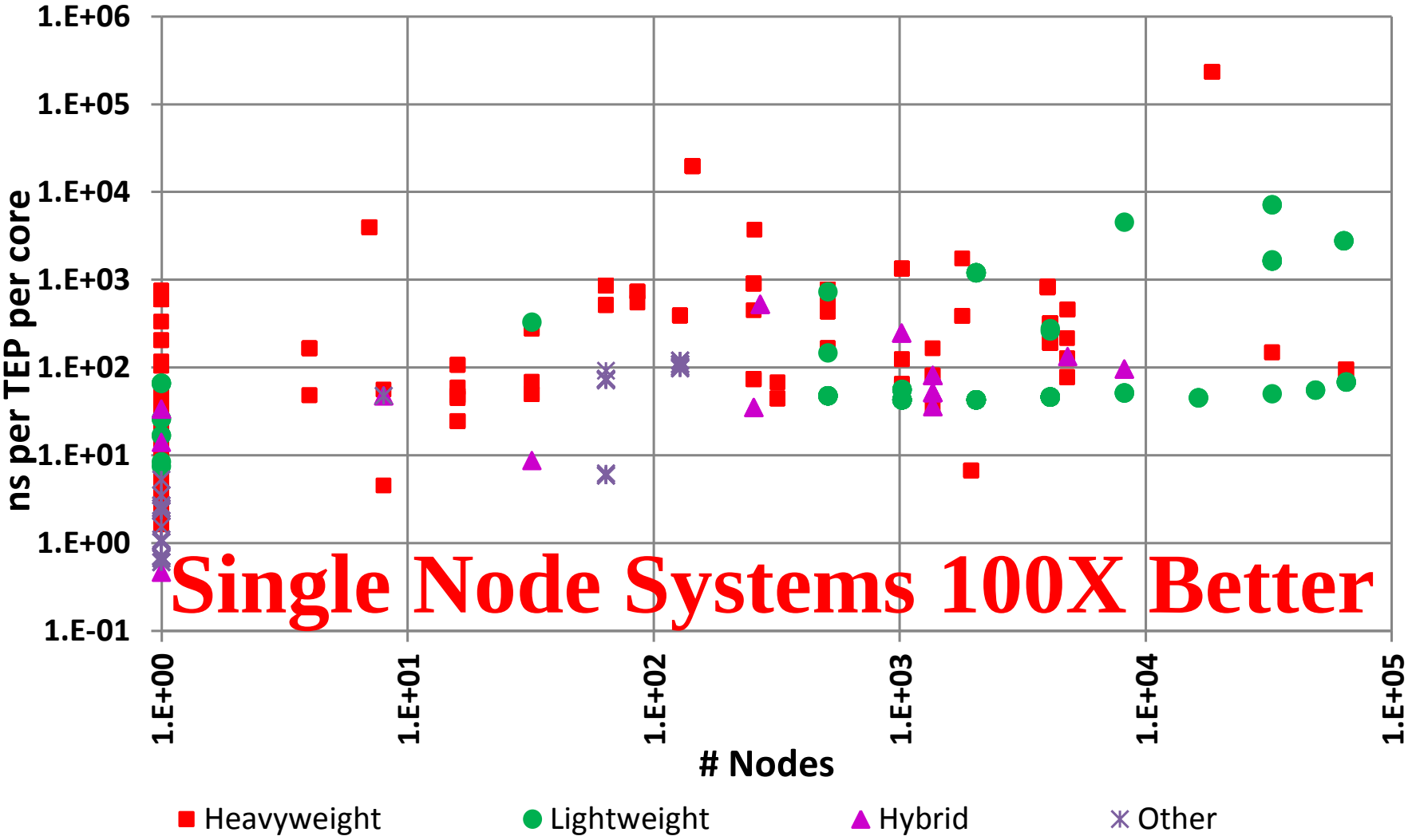
Where Does Improvement Come From? BlueGene Analysis



TEPS/Node vs Node Count

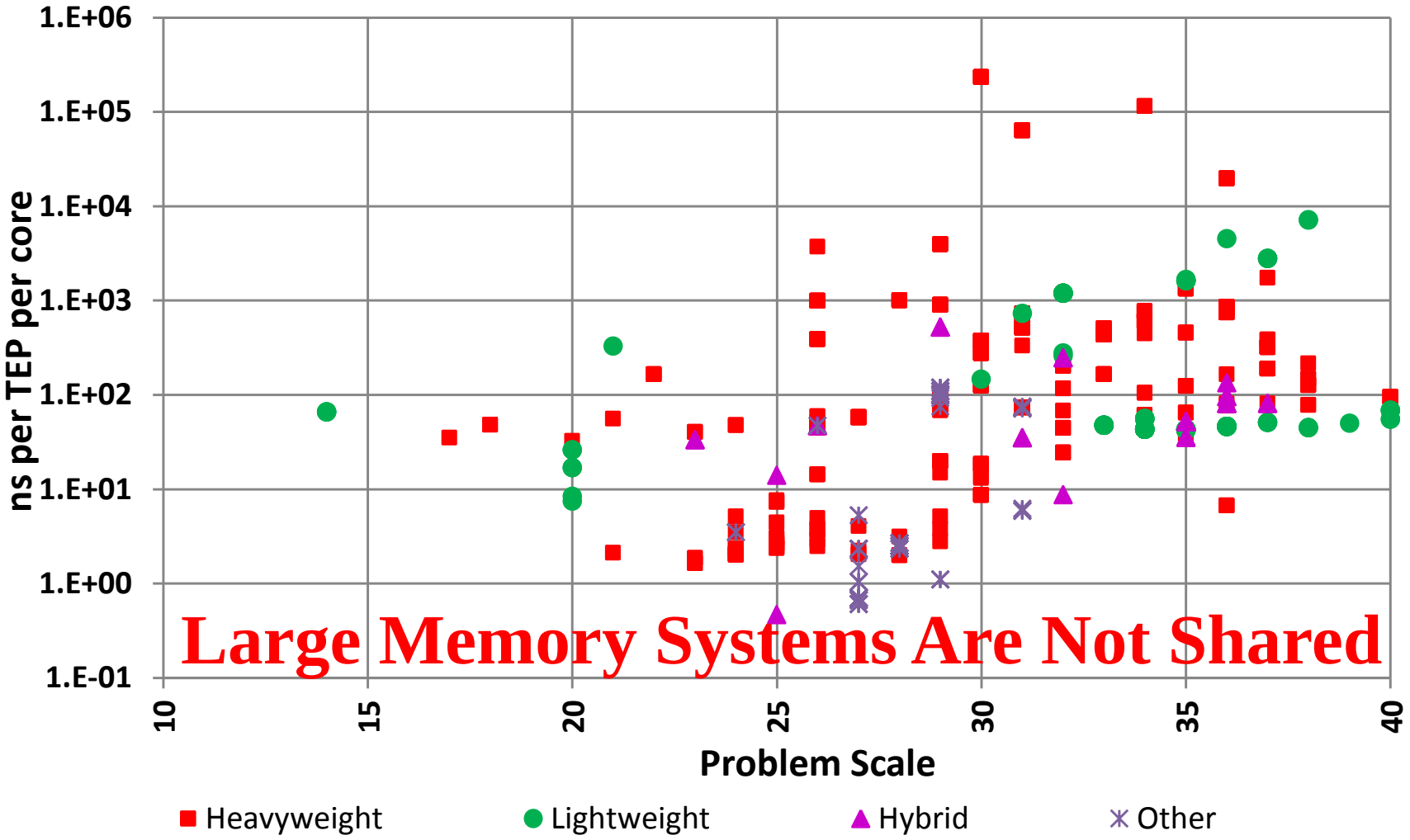


ns per TEP vs # Nodes



Single Node Systems 100X Better

ns per TEP vs Memory



Conclusions

- GRAPH500 has two metrics: TEPS & problem size
  - With problem size growing towards petabytes
- Algorithm: heavy on
  - remote atomic accesses
  - All to all comm patterns
- Near perfect weak scaling, but lose 100X in going from shared memory to distributed
- Comparative BlueGene data: ½ hardware, ½ algorithm (driven by new architectural features)
- New benchmarks coming: search, shortest path, max independent set

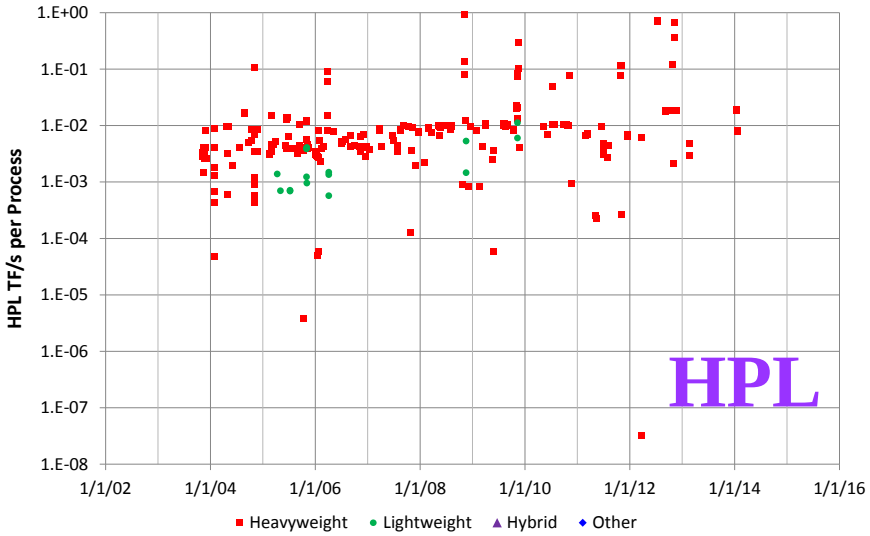
Topics

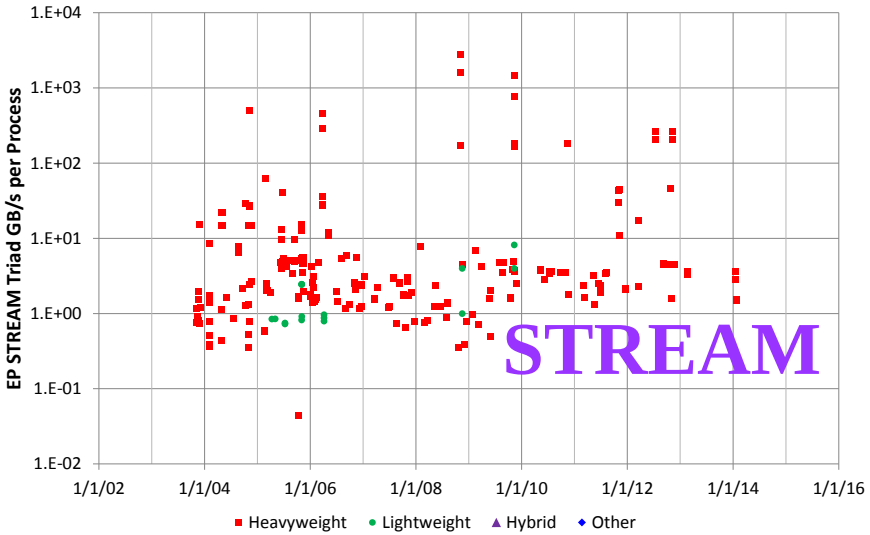
- 2004: Technology "Perfect Storm"
- Our Current Architecture Spectrum
- The Exascale Memory Energy Horror Show
- TOP500 Lessons
- GRAPH500 Lessons
- **HPCC Lessons**
- 3D Stack Projections

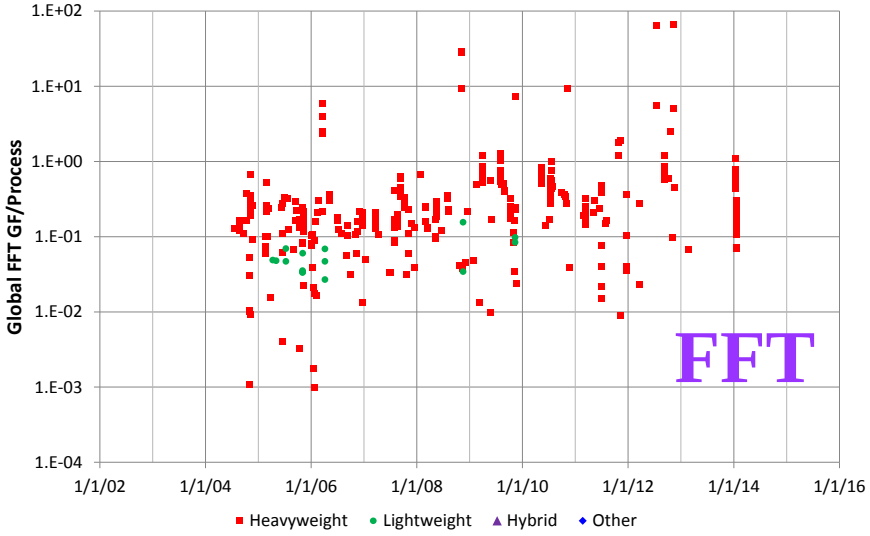
The HPCC Challenge Benchmarks

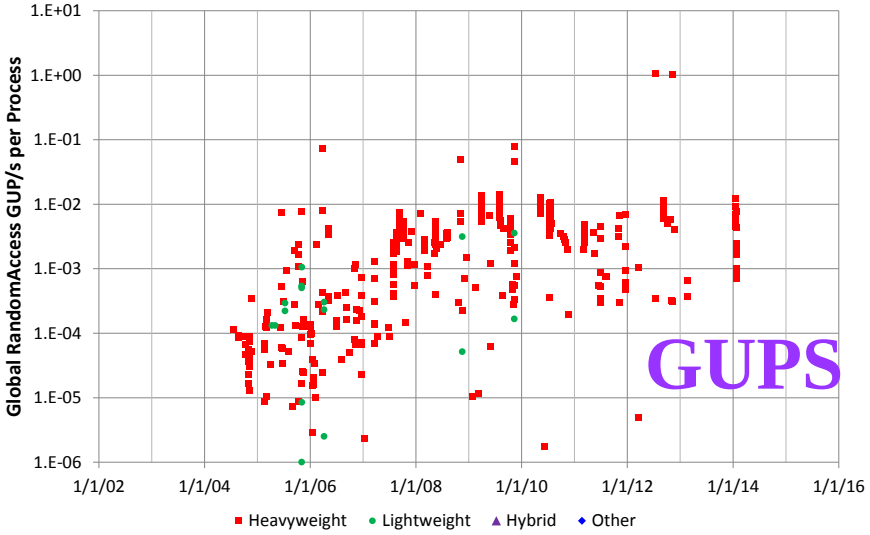
- Outgrowth of the DARPA HPCS program
- Focus on large-scale parallel MPI implementations
- 362 system measurements reported since 2003
- Benchmarks:
  - **HPL***: the LINPACK benchmark used in TOP500
  - **DGEMM**: DP FP matrix-matrix multiply
  - **STREAM***: Sustainable memory B/W for 4 long vector ops
    - TRIAD: a[i] = b[i] + c[i]*SCALAR
  - **PTRANS**: Comm capabilities for parallel matrix transpose
  - **RandomAccess***: Integer random updates (GUPS)
  - **FFT***: DP FP
- *ed versions reported at HPC Challenge Awards at SC

Performance/Process vs Time

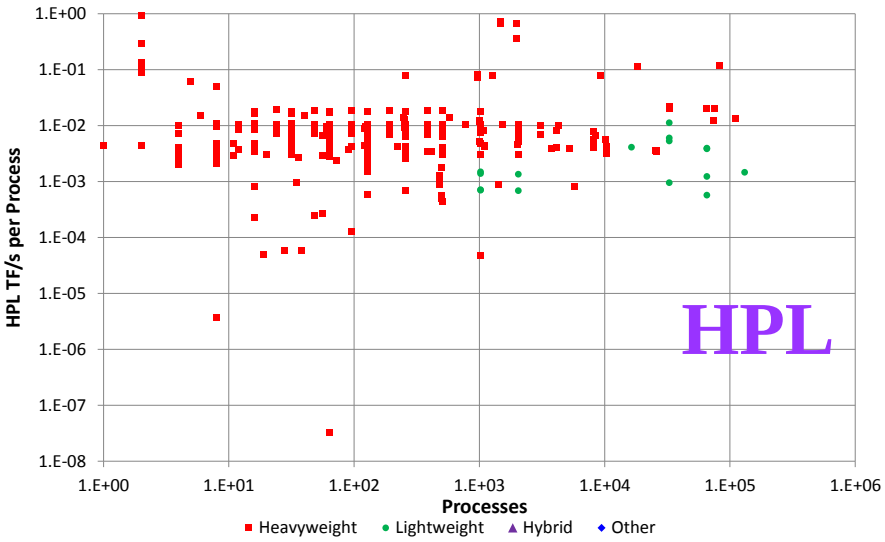


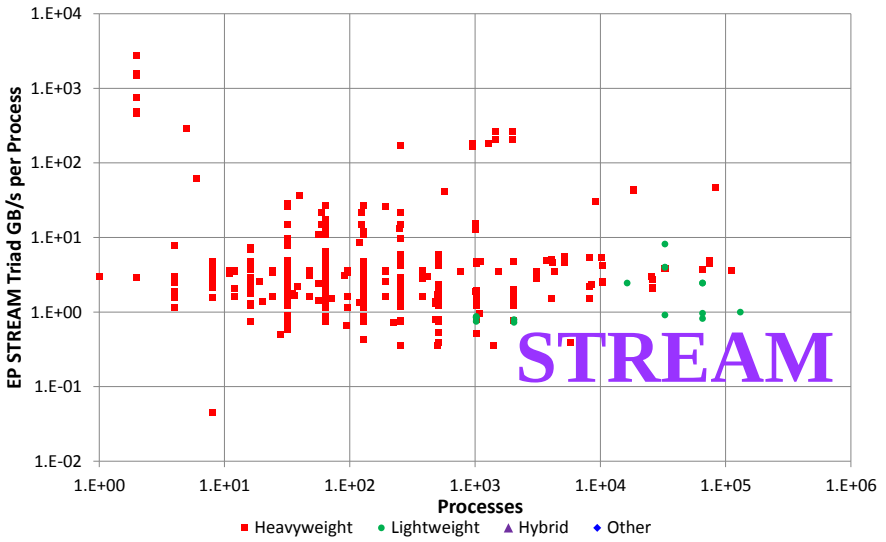


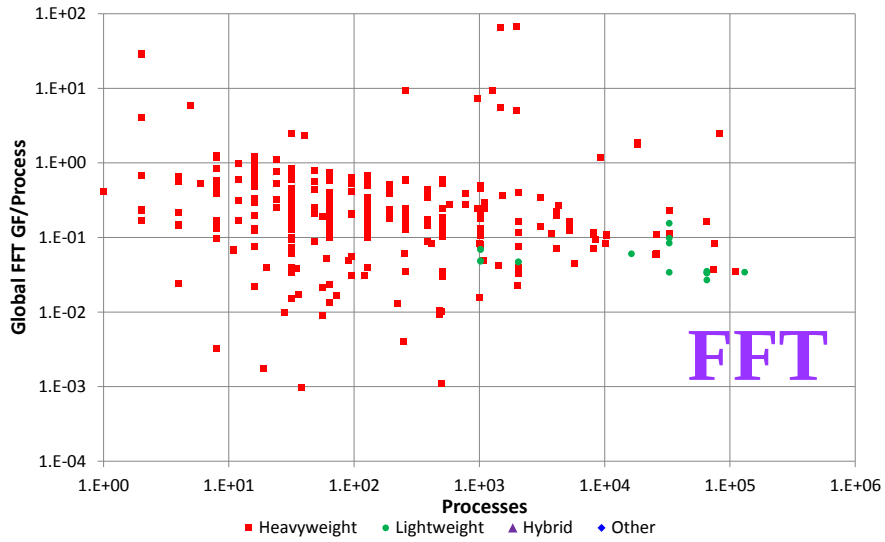


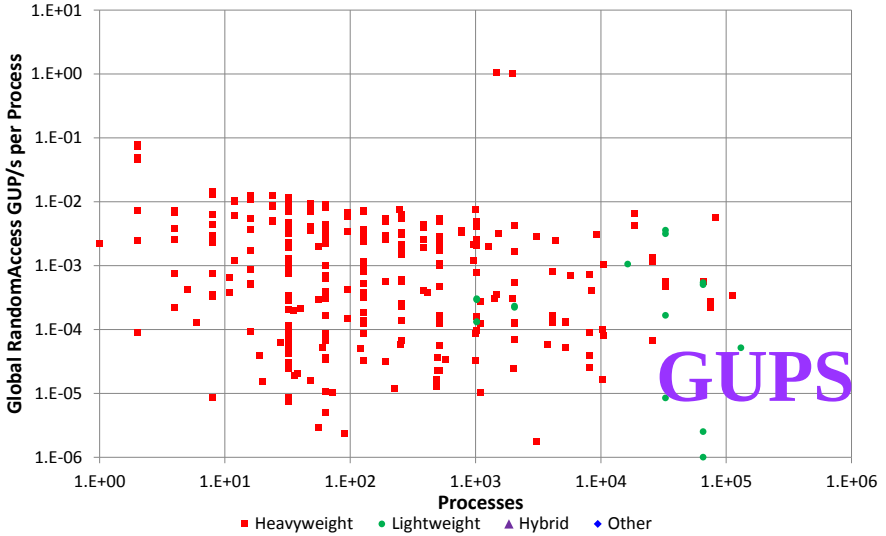


Performance/Process vs # Processes

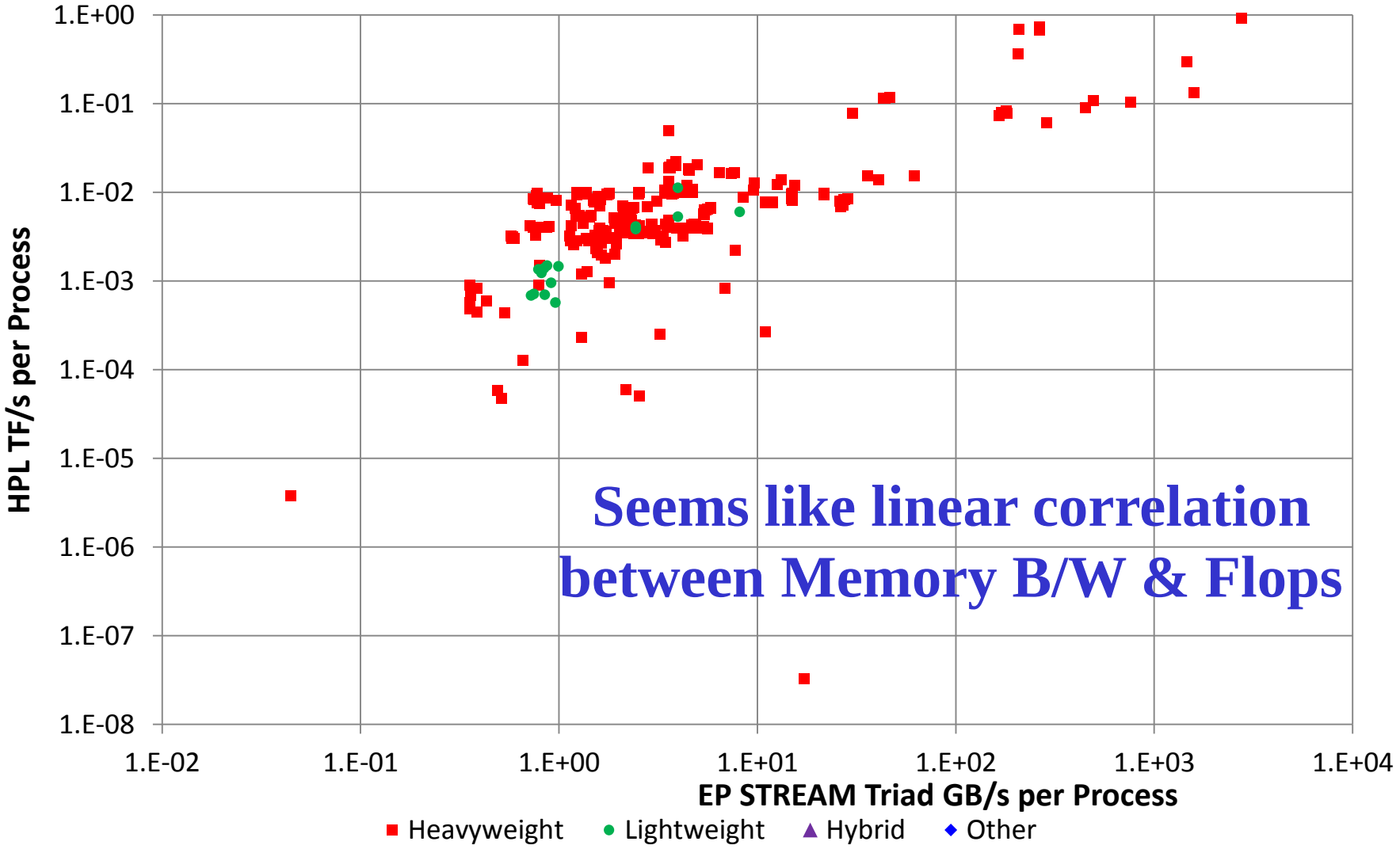




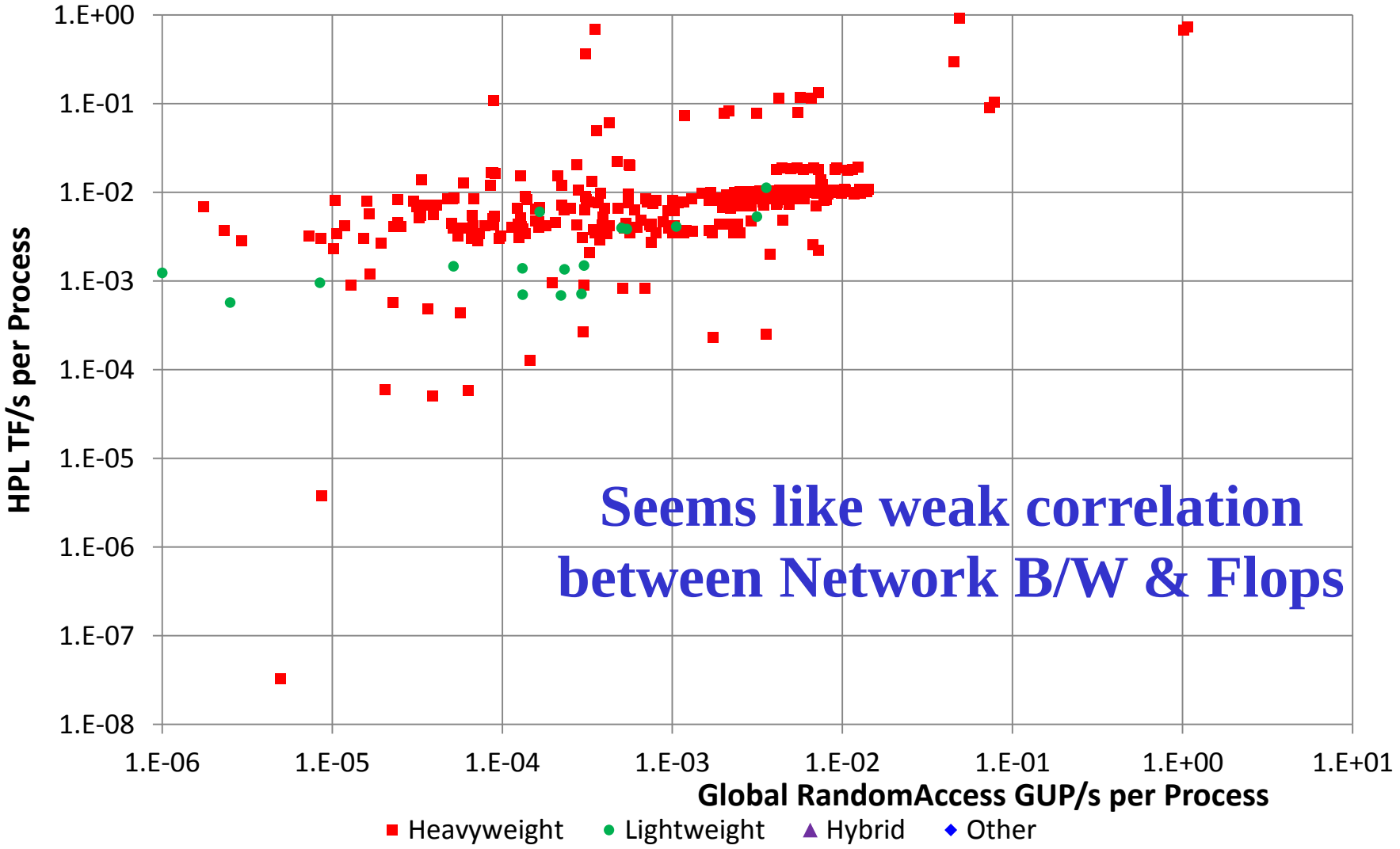




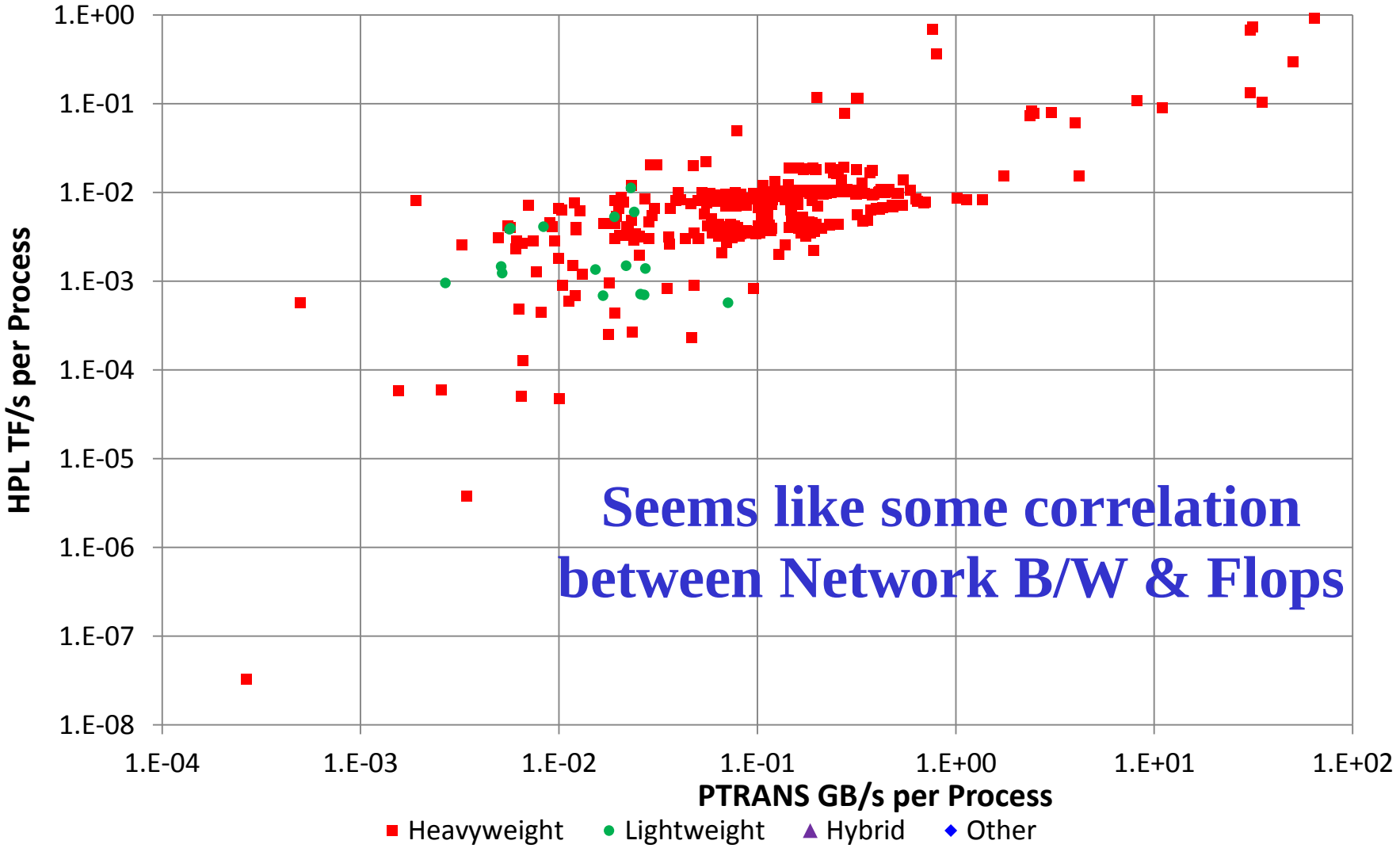
How Does HPL Relate to Streams?



How Does HPL Relate to Interconnect?



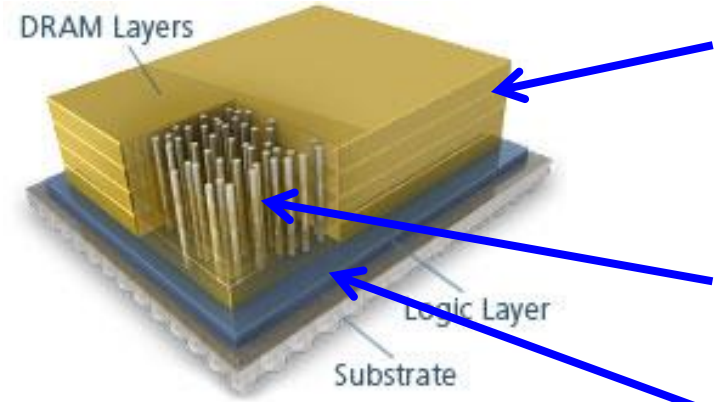
Another View Using PTRANS



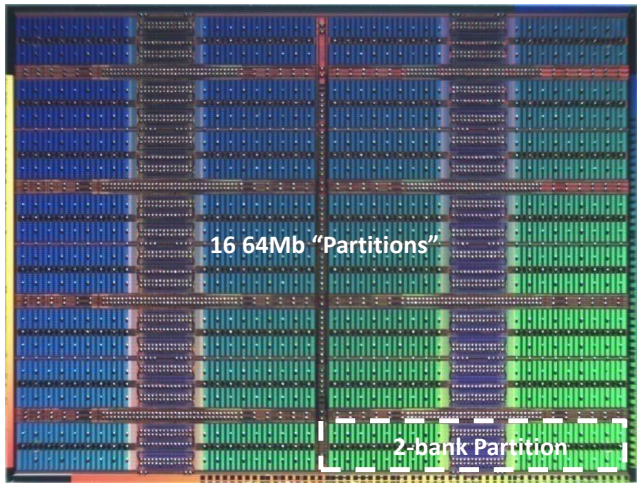
Topics

- 2004: Technology "Perfect Storm"
- Our Current Architecture Spectrum
- The Exascale Memory Energy Horror Show
- TOP500 Lessons
- GRAPH500 Lessons
- HPCC Lessons
- **3D Stack Projections**

Emerging Technology: 3D Stacks

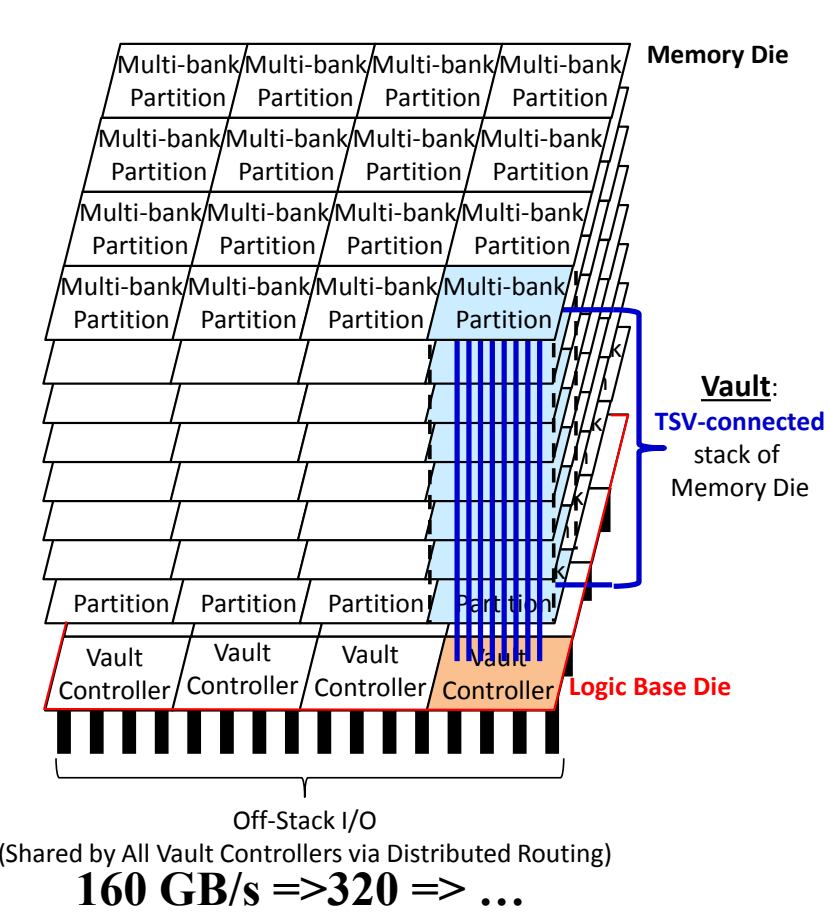


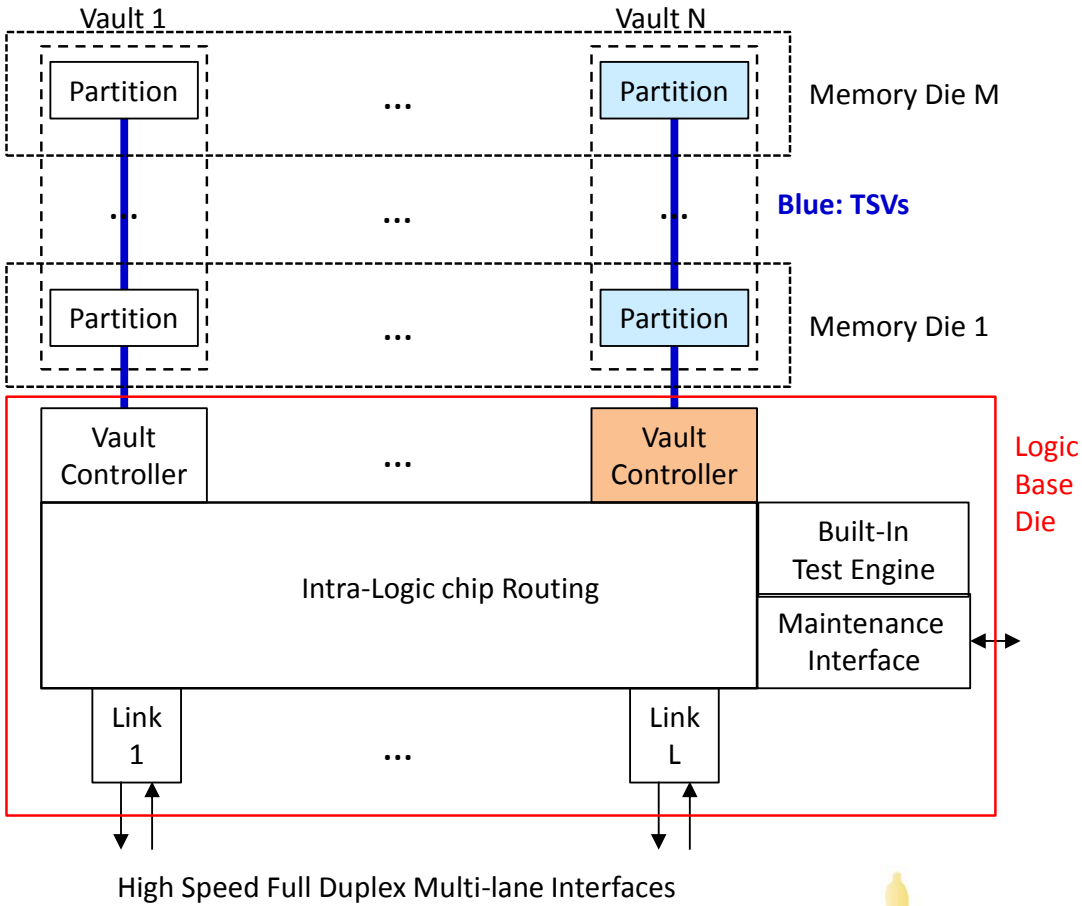
http://www.micron.com/products/hybrid-memory-cube



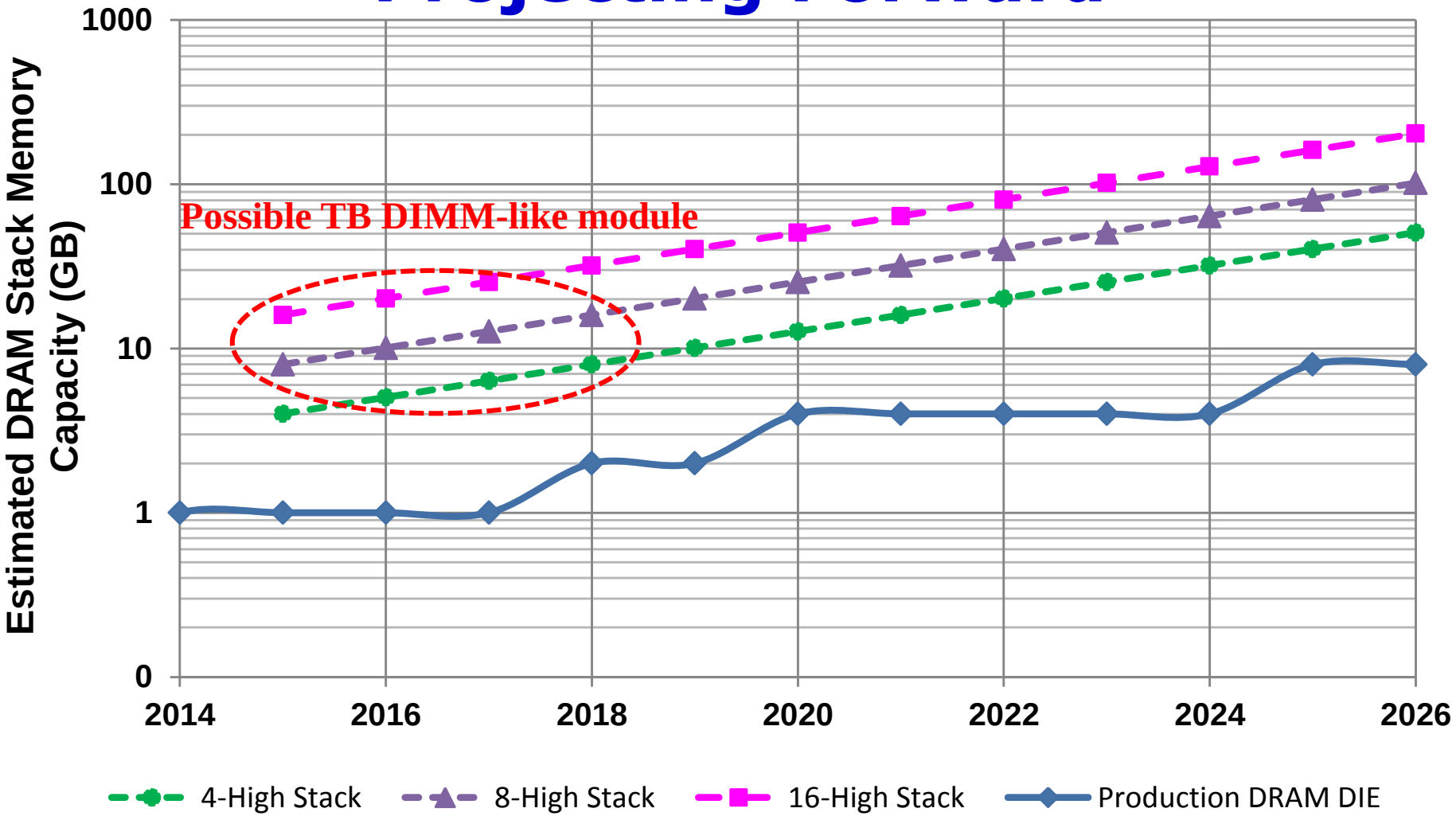
- Stackable memory chips
  - Same memory cells as in today's DRAM
  - Internally re-architected for more simultaneous access
- "Through Silicon Vias" (TSVs) to go vertically up and down
- Logic chip on bottom
  - Multiple memory controllers
  - More sophisticated off-stack interfaces than DDR
- Prototype demonstrated in 2011
- 1st Product in 2014
  - Spec:http://www.hybridmemorycube.org
  - Capacity: up to 8GB: 8X single chip
  - Bandwidth: up to 480GB/s: 40X
  - Lots of room on logic chip

The HMC Architecture





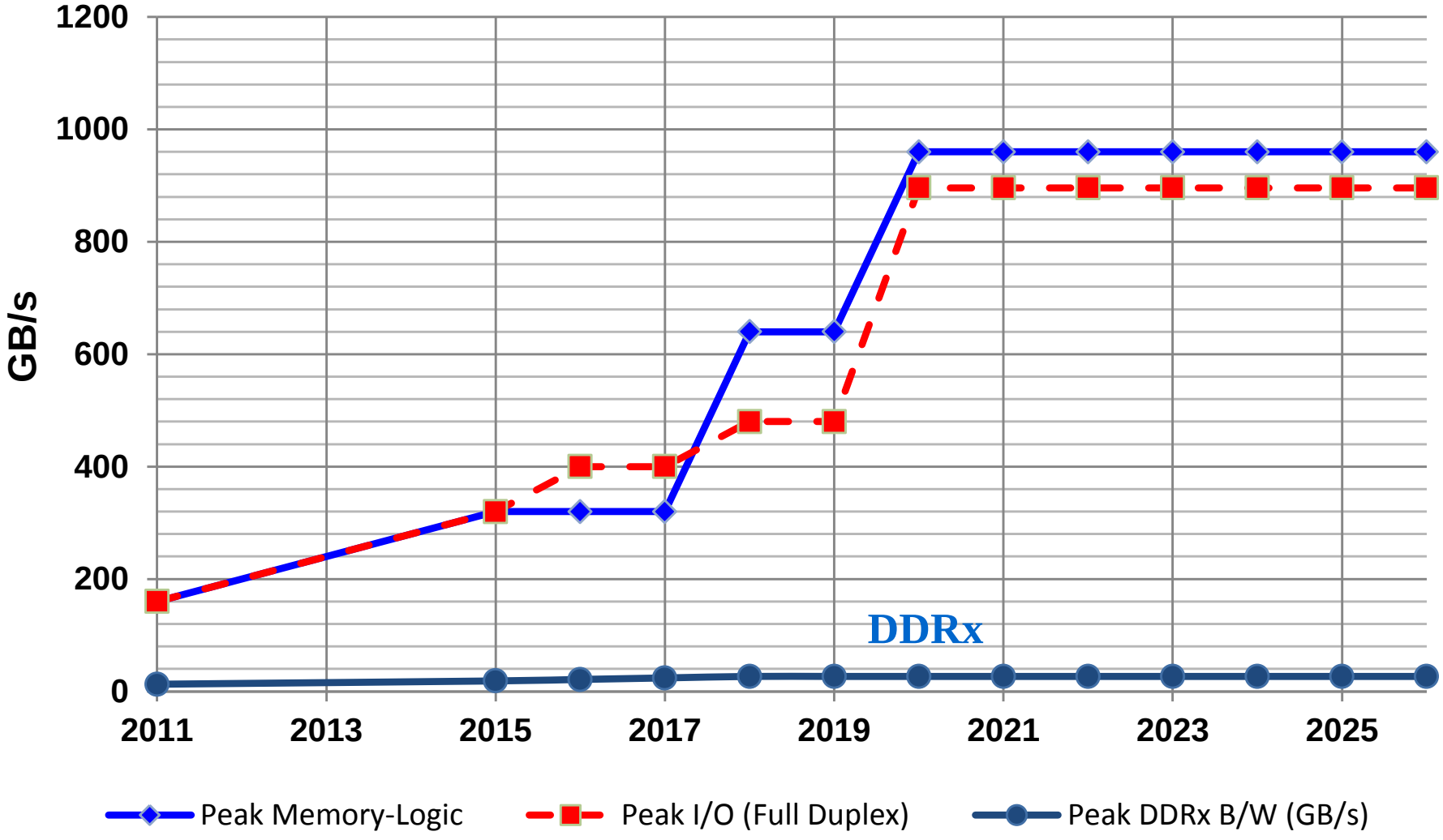
Projecting Forward



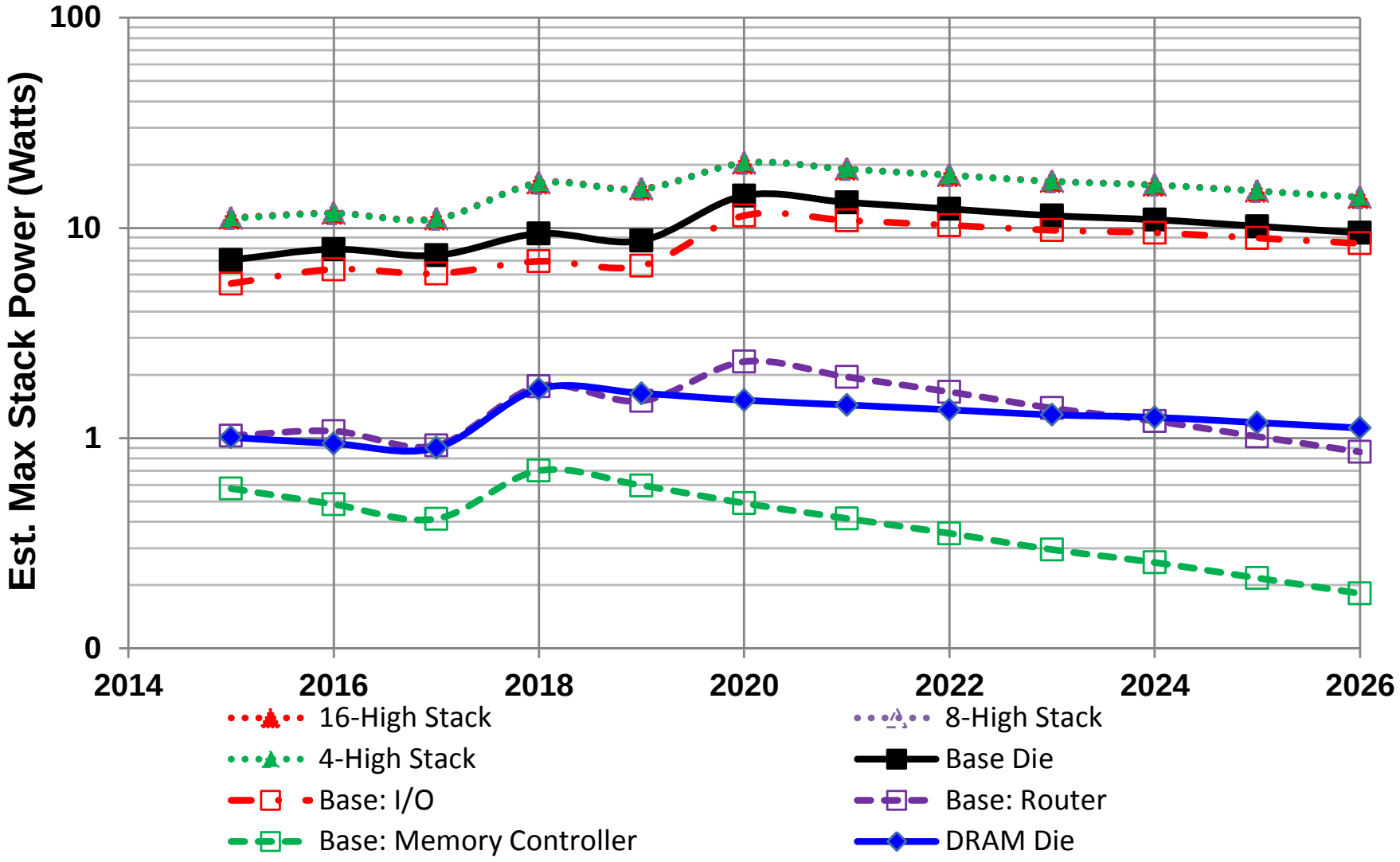
Remember: Stack takes ~ area of a single DRAM die.

Leading edge DDR DIMMs have up to 128 die.

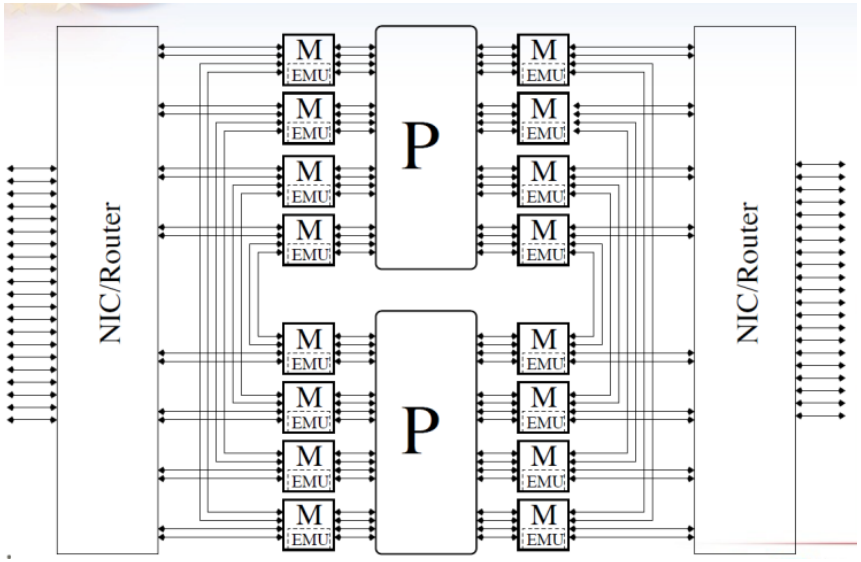
Bandwidth Estimates



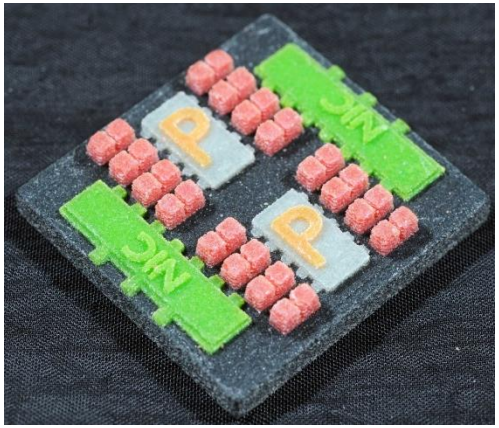
Stack Power Projections



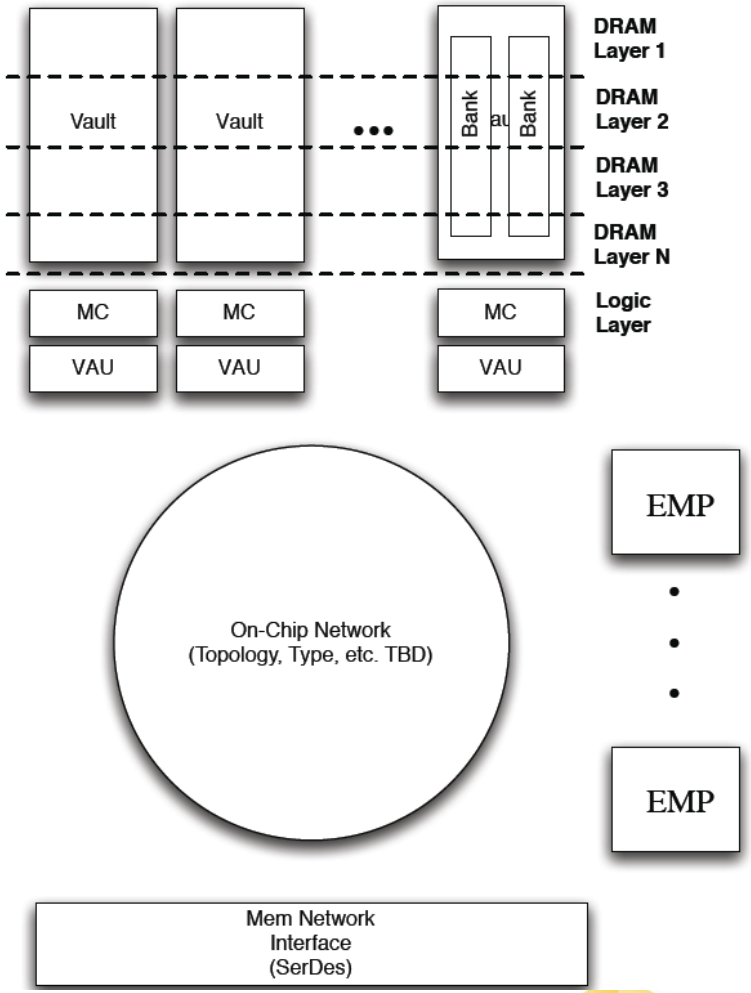
SNL Xcaliber Architecture



(a) X-caliber Node Architecture



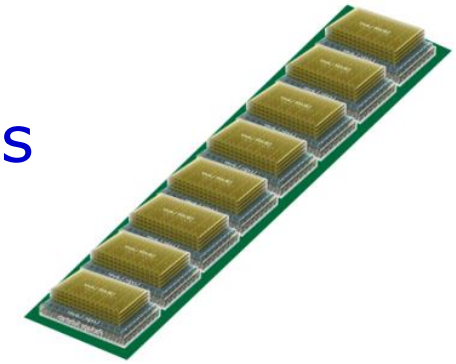
(b) X-caliber Node Mockup

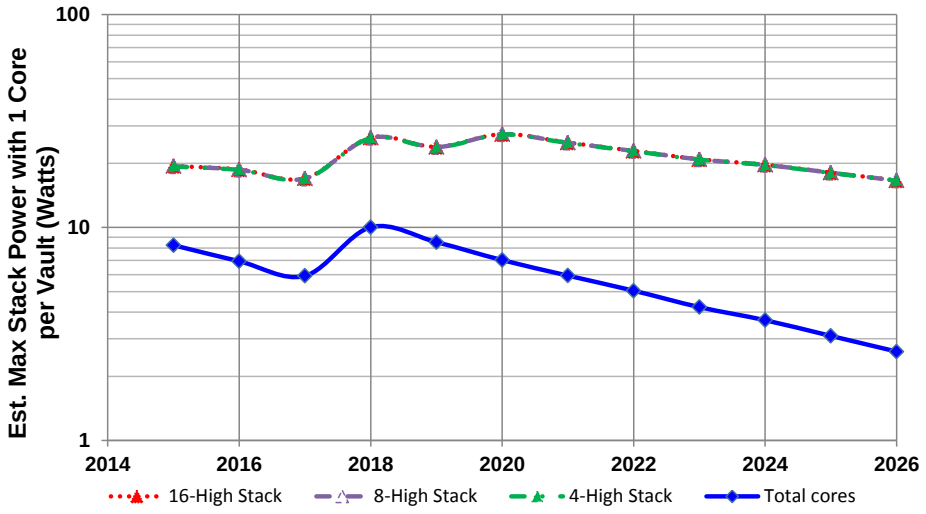


(c) X-caliber stack notional architecture

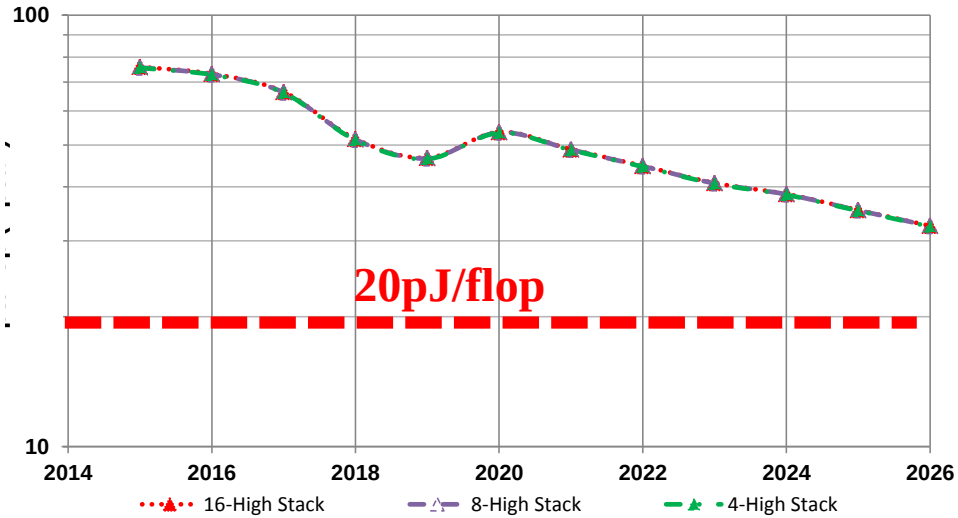
Thought Experiment: Extrapolate from HMC to "CMC"

- Add simple core + atomics/vault
- Ditch the Ps, NICs' and use only stacks
- Package multiple stacks per "DIMM"





Assumes all resources running 100%



If off-chip and memory bandwidth <100% we are almost there!

Possible 3D Architectures

- Advantages
  - Already has significant internal resiliency
  - Huge bandwidth to logic chip
  - Huge off-stack bandwidth with better protocol
  - Very much in line with lightweight architectures
- Future Possibilities
  - Direct mount multiple stacks to same substrate
  - Lightweight cores + atomics natural per vault
  - Heterogeneous memory die in stack to provide on die checkpointing, storage, etc.
  - Multitude of novel resiliency options
  - Augment protocol to support active messages
- Study of a stack-based Big Data solution: Huge increase in performance & performance/unit volume

Overall Summary

- 2004: Turning point in HPC architecture
- Heavyweight alone are going flat
- Hybrid more energy efficient for flops – but there are memory issues
- Lightweight more scalable for both flops and communication
- No escape for massive parallelism
- New algorithms will blend multiple metrics
- 3D stacks an interesting alternative