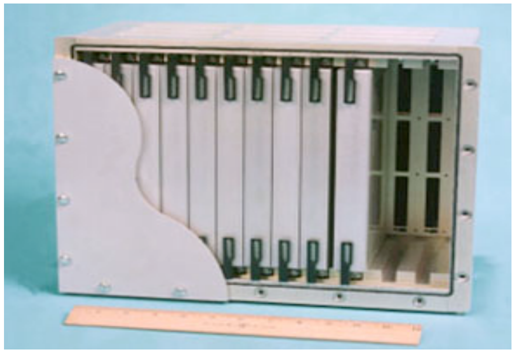
The Challenges of Exascale Computing for Astroinformatics Apps



TeraFlop
Embedded

PetaFlop
Departmental

ExaFlop
Data Center

Peter M. Kogge
McCourtney Chair in Computer Science & Engineering
Univ. of Notre Dame
IBM Fellow (retired)

June 16, 2010

UNIVERSITY OF NOTRE DAME

Topics

- Exascale and Astroinformatics
- Moore's Law, Multicore, & Energy 101
- The DARPA Exascale Study
- Exascale Strawmen
- A Deep Dive into Memory Energies

Materials presented here based on an 18 month DARPA-sponsored Study on "Exascale Computing"

Background-Computing

- **Classical "Measures of Performance"**
  - ***Flop/s***: floating point operations per second
  - ***Bytes per Flop/s***: memory capacity per unit of performance
    - Classical Goal: 1 byte per flop/s
  - **Bytes/s per Flop/s**: memory bandwidth per unit of performance
    - Classical Goal: 1 byte/s per flop/s

- **Last 30 years of HPC: focus on flops/s**
  - 1980s: "Gigaflop" (10^9/s) first in single vector processor
  - 1994: "Teraflop" (10^{12}/s) first via thousands of microprocessors
  - 2009: "Petaflop" (10^{15}/s) first via several hundred thousand cores

- **Let's call *technology* to reach "xyz flop" as "*xyz scale*"**

Background-Technology

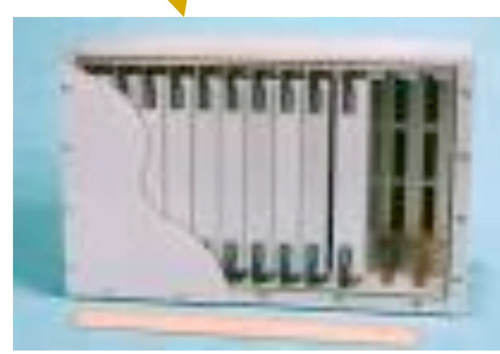
- **Commercial technology: *to date***
  - Riding Moore's Law to denser, "faster," chips
  - Always shrunk prior "XXX" scale systems to smaller form factor
  - Shrink, with clock speedup, enabled next "XXX" scale
- **But microprocessor clock rates flat-lined in 2004**
  - Driven by power dissipation limits
- **"Exascale" now on horizon (Exaflops = 10^{18}/s)**
  - But technology change radically changing our path to get there

Especially Energy/Power

- **And very relevant to Astroinformatics**

Exascale

- **Exascale = *1,000X* *capability* of Today**
- **Exascale != Exaflops but**
  - Exascale at the data center size => ***Exaflops***
  - Exascale at the “rack” size => ***Petaflops*** for departmental systems
  - Exascale embedded => ***Teraflops*** in a cube
- **It took us 14+ years to get from**
  - 1st Petaflops workshop: 1994
  - Thru NSF studies, HTMT, HPCS …
  - To give us to Petaflops *in 2009*
- **Study Questions:**
  - Can we ride silicon to Exa?
  - What will such systems look like?
  - Where are the Challenges?



For Those of You Who Want an Exaflop…..

You should be OVERJOYED if all you will need is:

- ***JUST a Million cores***
- ***ONLY 1 Nuclear Power Plant***
- ***MINIMAL programming support***

Astroinformatics: Data Intensive

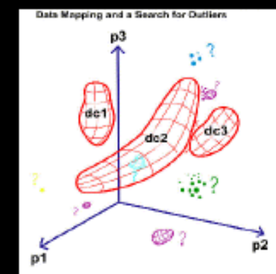
From Data-Driven to Data-Intensive

- Astronomy has always been a data-driven science
- It is now a data-intensive science: **Astroinformatics**
  - And it will become even more data-intensive in the coming decade(s)
- Some key data-driven questions for astronomers:
  - What is it?
  - Where is it?
  - What causes that behavior?
  - When did it form?
  - How did it form?
  - Why did it do that?
  - Who will let me use their telescope to get more data???

Knowledge !

Informatics = Data-Enabled Science:
Scientific KDD (Knowledge Discovery from Data)

- Characterize the known (clustering, unsupervised learning)
- Assign the new (classification, supervised learning)
- Discover the unknown (outlier detection, semi-supervised learning)

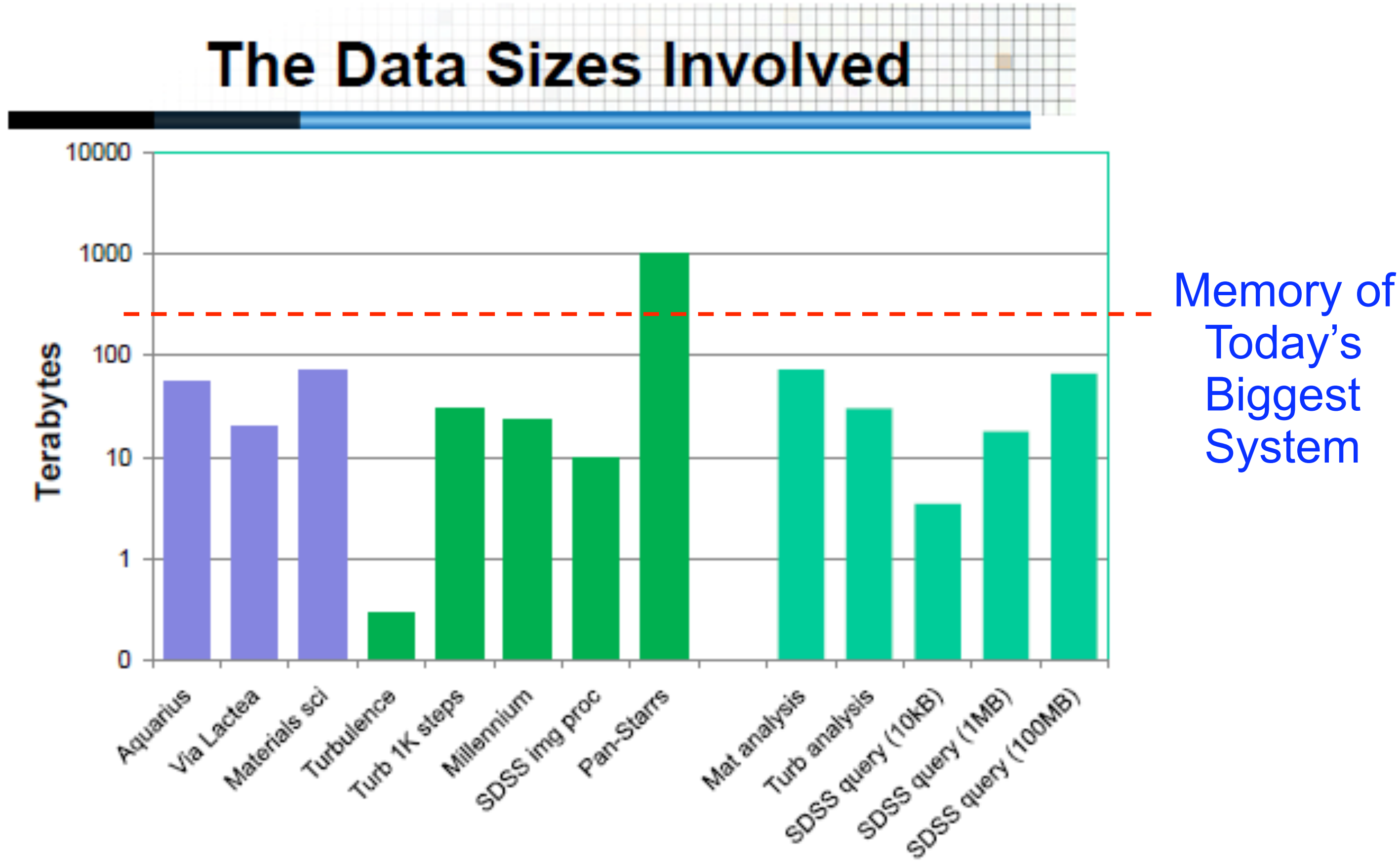


Graphic from S. G. Djorgovski

- **Benefits of very large datasets:**
  - **best statistical analysis of "typical" events**
  - **automated search for "rare" events**

Slides 5 & 6 from Kirk Borne,
"Scalable Peer-to-Peer Data Mining for Data-Intensive Astroinformatics,"
2010 Salishan Conf. on High Speed Computing

Today's Astro Data Sets Are Large … And Growing



From Alex Szalay, " Amdahl's Law and Extreme Data-Intensive Computing," 2010 Salishan Conf. on High Speed Computing

Growth rates can exceed 1 PB/year for Raw Data - LSST may reach 100 PB!

And What We Want to Do With It Often Requires Faster than Disk Access!!

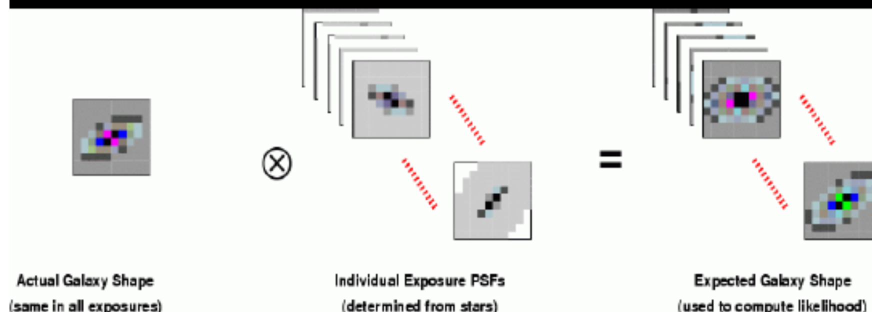
Some key astronomy problems that require informatics and data science techniques

- Probabilistic Cross-Matching of objects from different catalogues
- The distance problem (*e.g.,* Photometric Redshift estimators)
- Star-Galaxy Separation
- Cosmic-Ray Detection in images
- Supernova Detection and Classification
- Morphological Classification (galaxies, AGN, gravitational lenses, ...)
- Class and Subclass Discovery (brown dwarfs, methane dwarfs, ...)
- Dimension Reduction = Correlation Discovery
- Learning Rules for improved classifiers
- Classification of massive data streams
- Real-time Classification of Astronomical Events
- Clustering of massive data collections
- Novelty, Anomaly, Outlier Detection in massive databases

Both data & computation Intensive

Example: Petascale Computational & Data Science Challenge problem for LSST

- Find the optimal simultaneous solution for 20,000,000,000 objects' shapes across 2000 image planes, each of which has 201x4096x4096 pixels ... 10^{23} floating-point operations!
  - This illustrates an example for just one such object:



References:
http://universe.ucdavis.edu/docs/MultiFit-ADASS.pdf
http://code.google.com/p/multifit/

With Problems approaching 10^5 Exaflops

Again from Kirk Borne,
"Scalable Peer-to-Peer Data Mining for Data-Intensive Astroinformatics,"
2010 Salishan Conf. on High Speed Computing

And Radio Astronomy Is Even Worse!

- **Example SKA: Square Kilometer Array**
  - Goal: create detailed RF spectral
  - 2000+ Dishes, each with multiple beams, delivering 8GB/s
  - FFTs on each beam to frequency bin data
  - N^2 cross correlations on each bin
  - ***Exaops/sec*** CONTINUOUSLY
- **No way to save all observations: long term integration to increase S/N**
  - Still have huge sets of data to archive
  - And then analyze as in optical
- **IN ADDITION: want to do real-time FAST TRANSIENT detection to capture interesting data sequences**
  - Close to Petascale problem in its own right

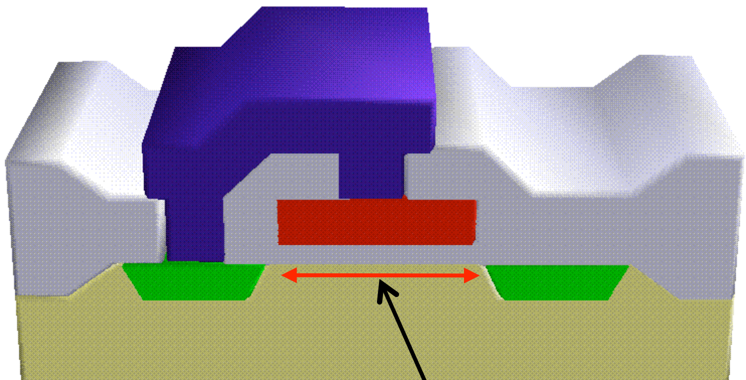
Supercomputers with Antennas!

Summary: Across the Board Need for Exascale Computing Technology

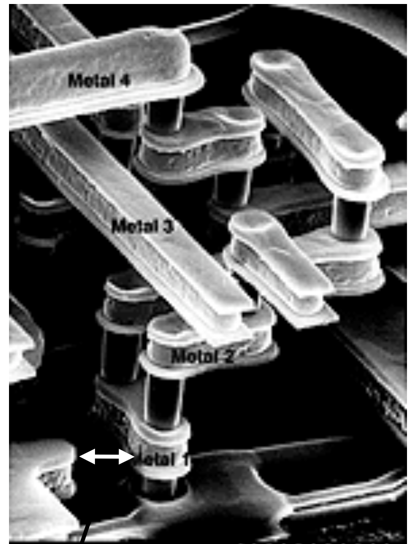
- **Tera Embedded: Sensor processing**
  - FFTs & Correlators for Radio
  - Replicated on massive scales
- **Peta Racks: Transient Event Detection**
- **Exa Data Centers: data mining**
  - May be physically distributed

Moore's Law, Energy 101, & Multicore

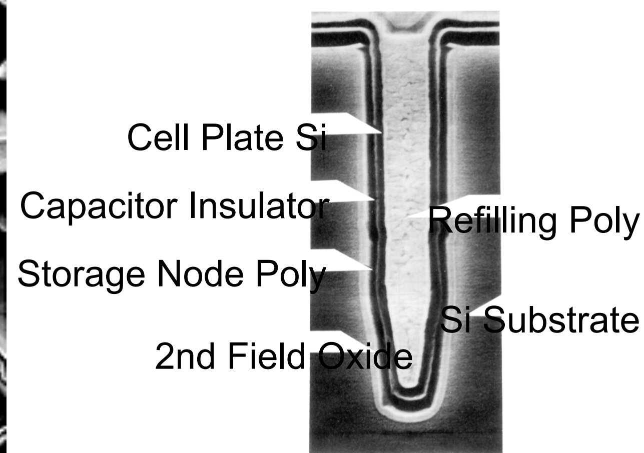
Today's CMOS Silicon Structures



A Single Transistor



Multiple Layers
Of Wire



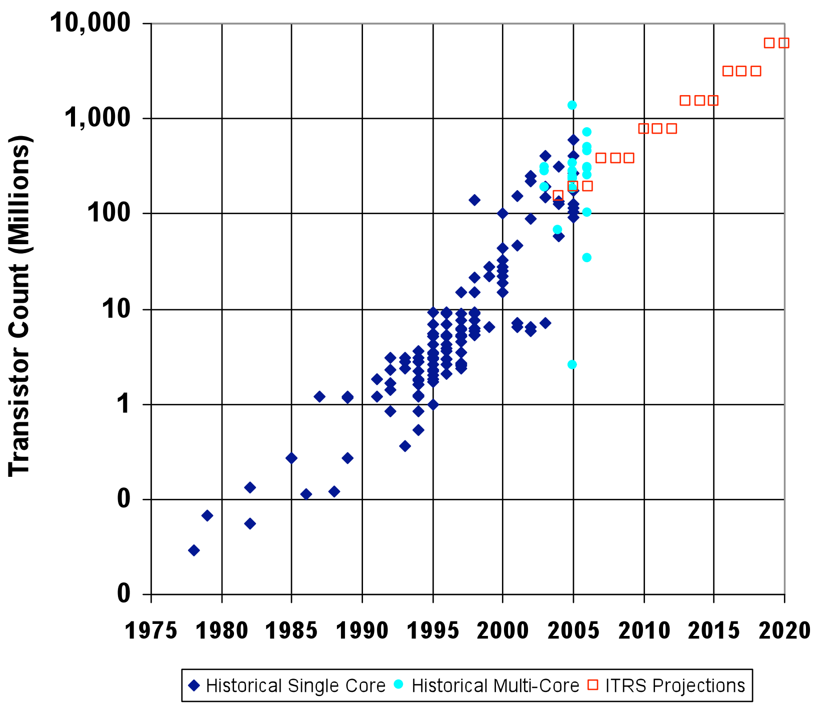
A Trench Capacitor
For Memory

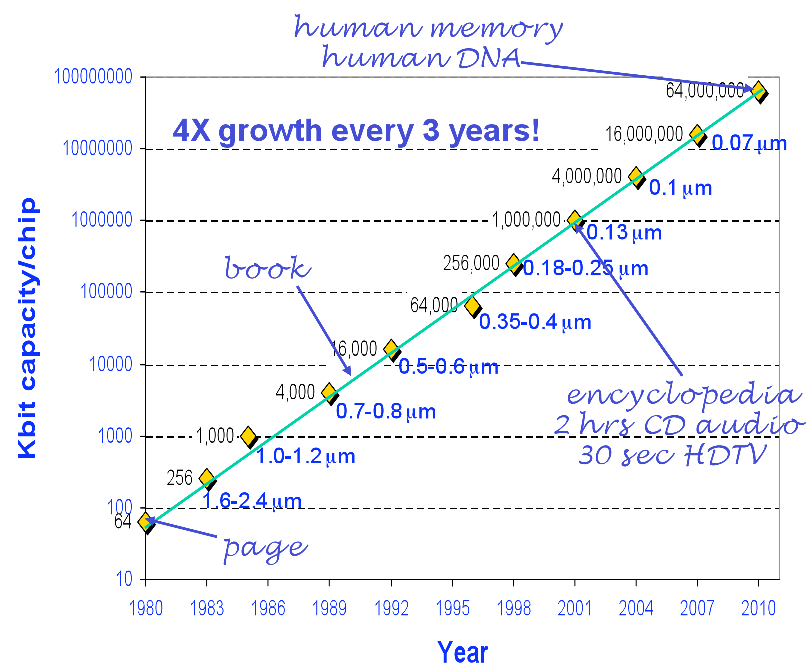
"Feature Size:" Minimum linear dimension

- Width of Transistor Gate
- ½ pitch between two wires at lowest levels of metal

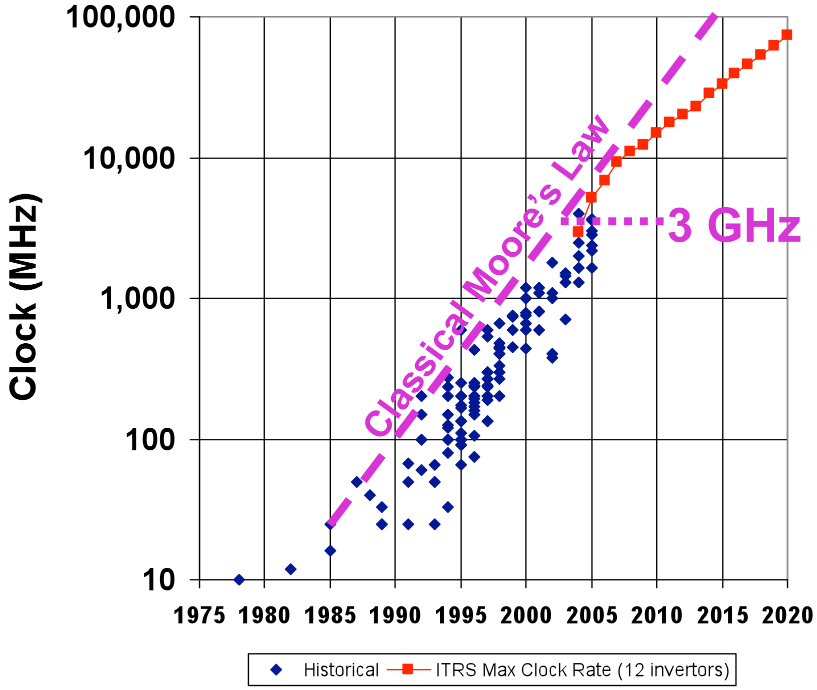
Moore's Law

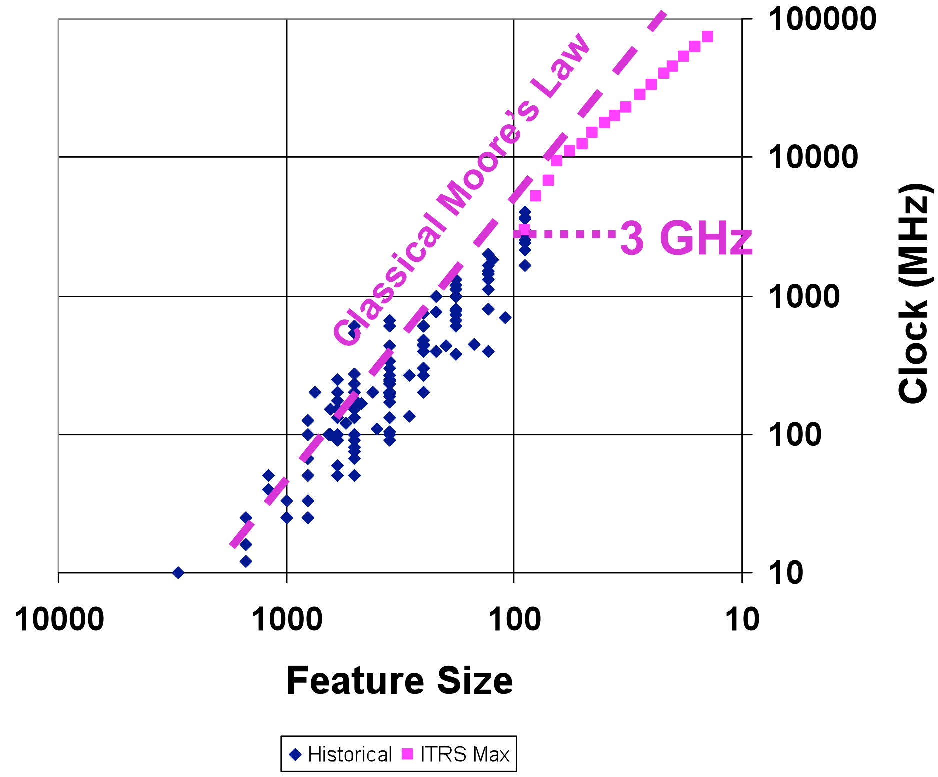
- **1965, Gordon Moore: "the number of transistors that can be integrated on a die would double every 18 to 14 months."**
  - Equivalent to Feature Size decreasing by factor of √2
  - Memory density 4X every 3 years





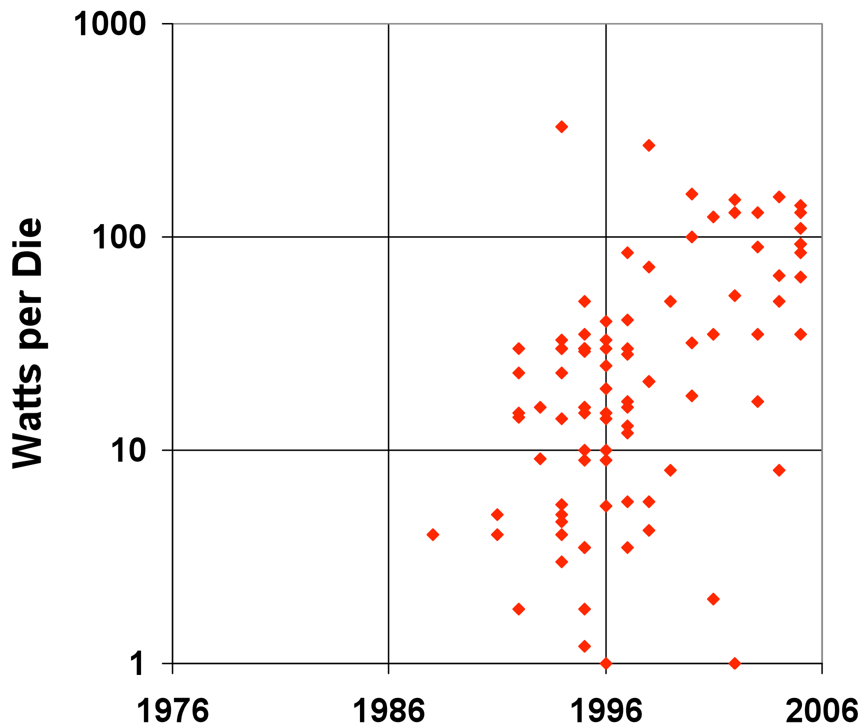
Smaller Feature Sizes Also Enable Higher Clock Rates

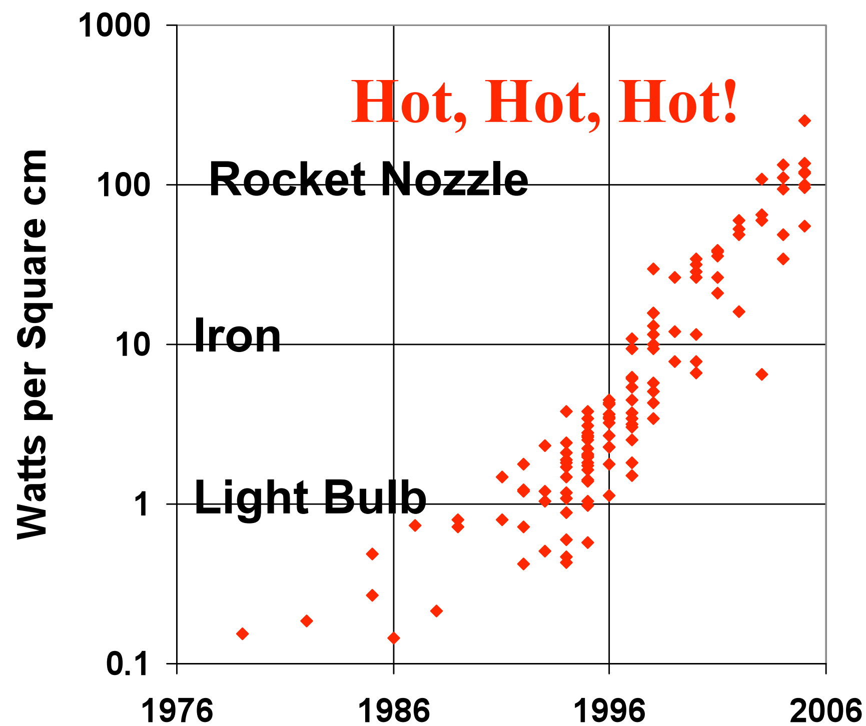




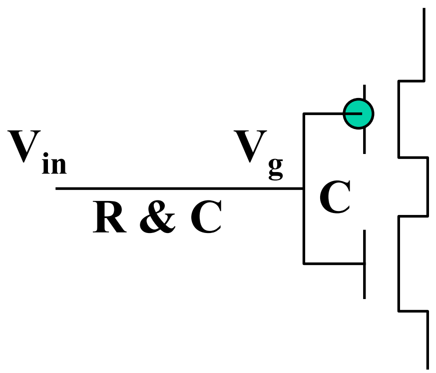
2005 projection was for 5.2 GHz – and we didn't make it. Further, we're still stuck at 3+GHz in production.

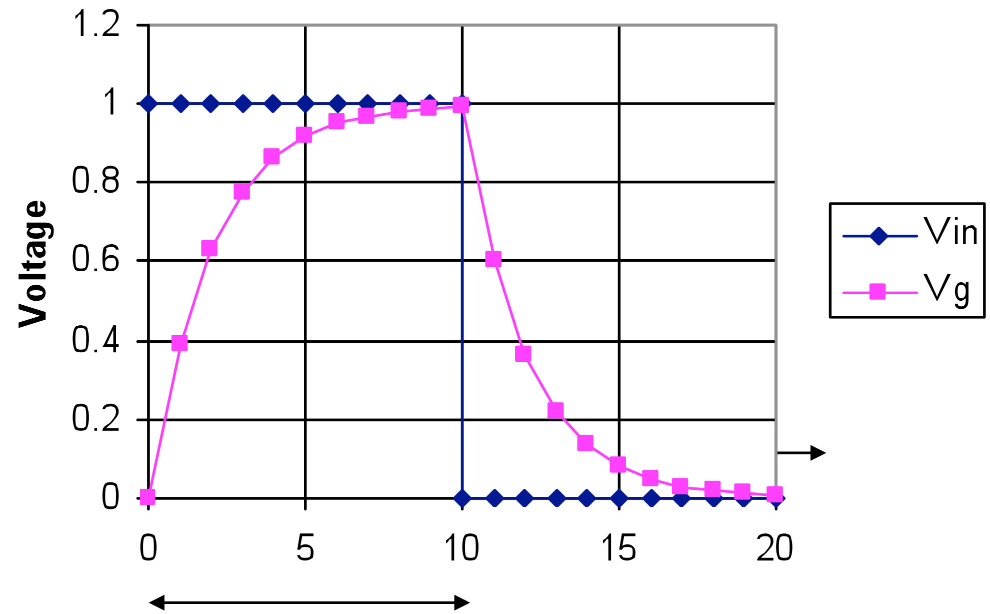
Why the Clock Flattening? *POWER*





CMOS Energy 101



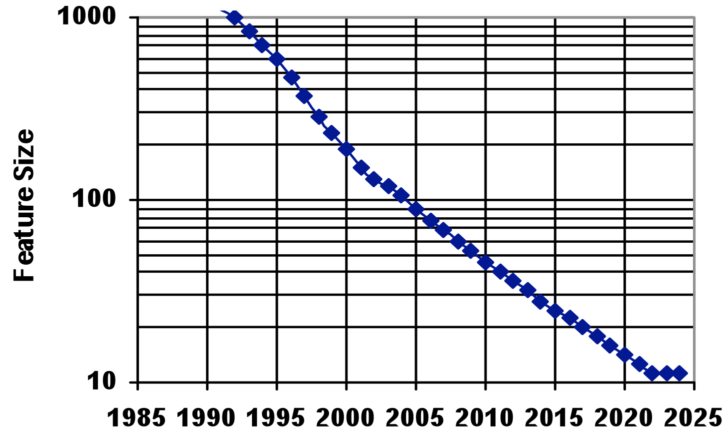


Dissipate $CV^2/2$ And store $CV^2/2$

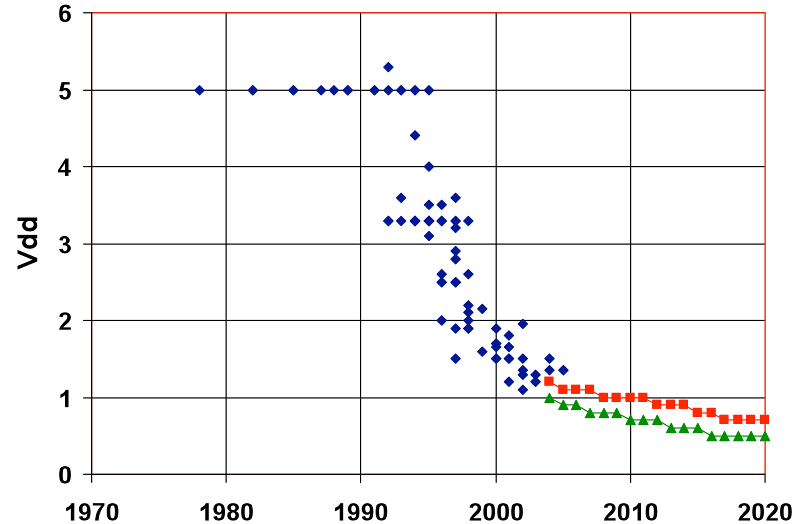
Dissipate $CV^2/2$ From Capacitance

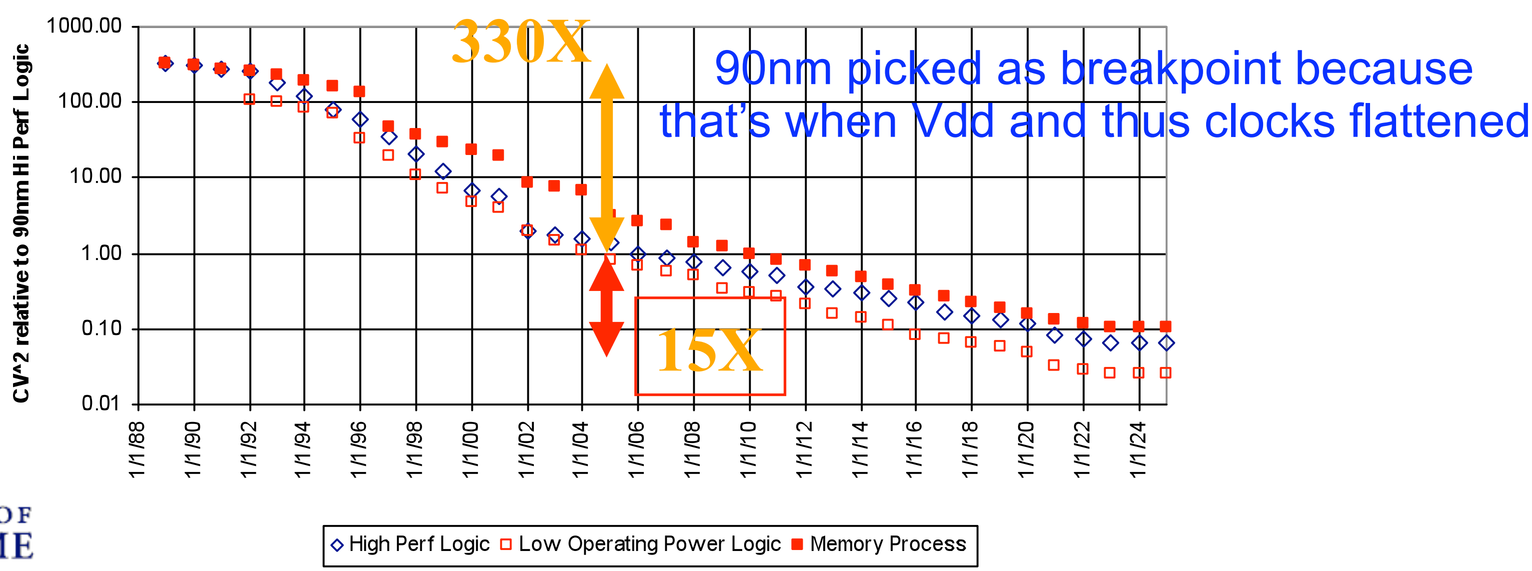
One clock cycle dissipates $C*V^2$

How Did CV^2 Improve With Time?



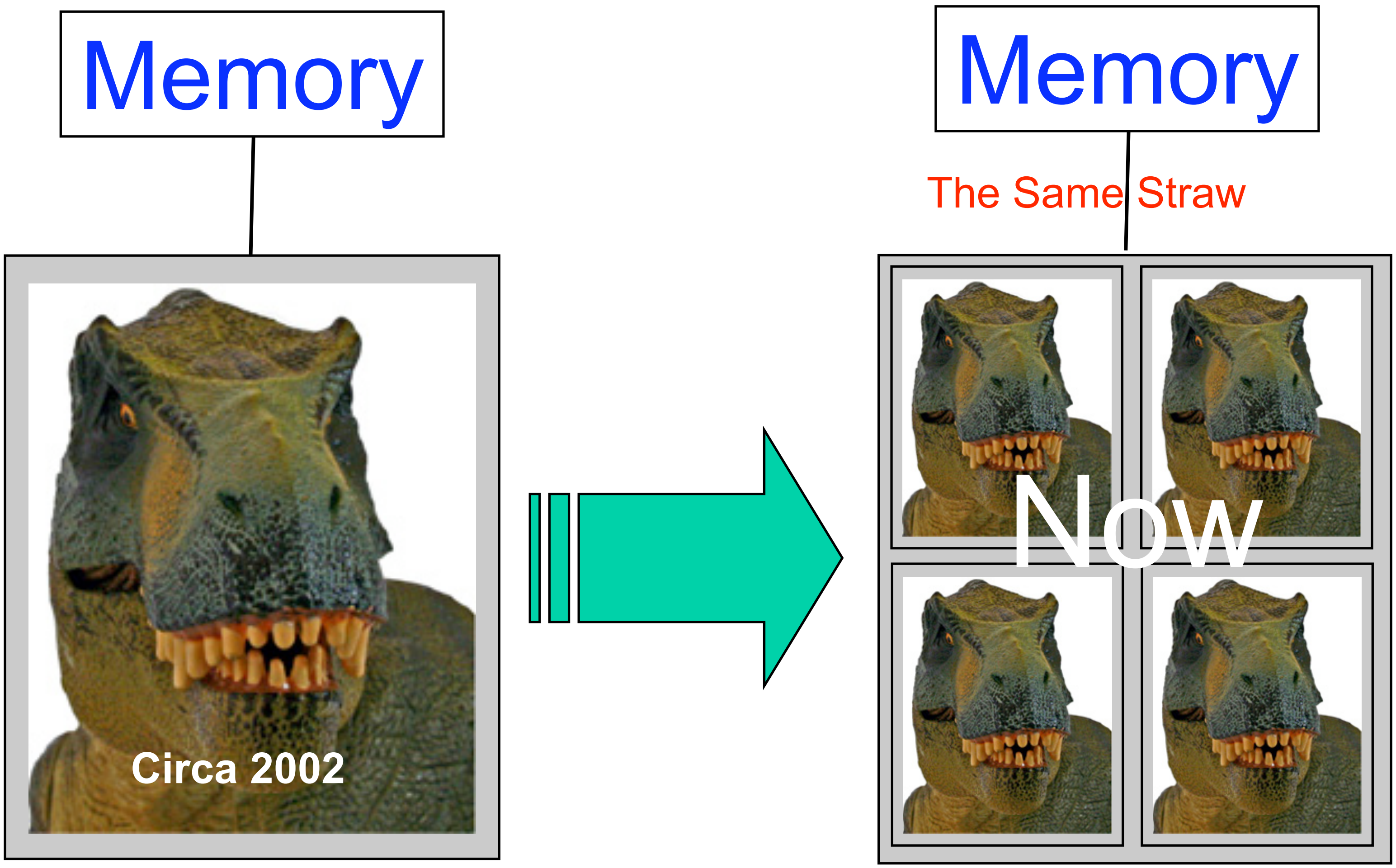
Assume capacitance of a circuit scales as feature size



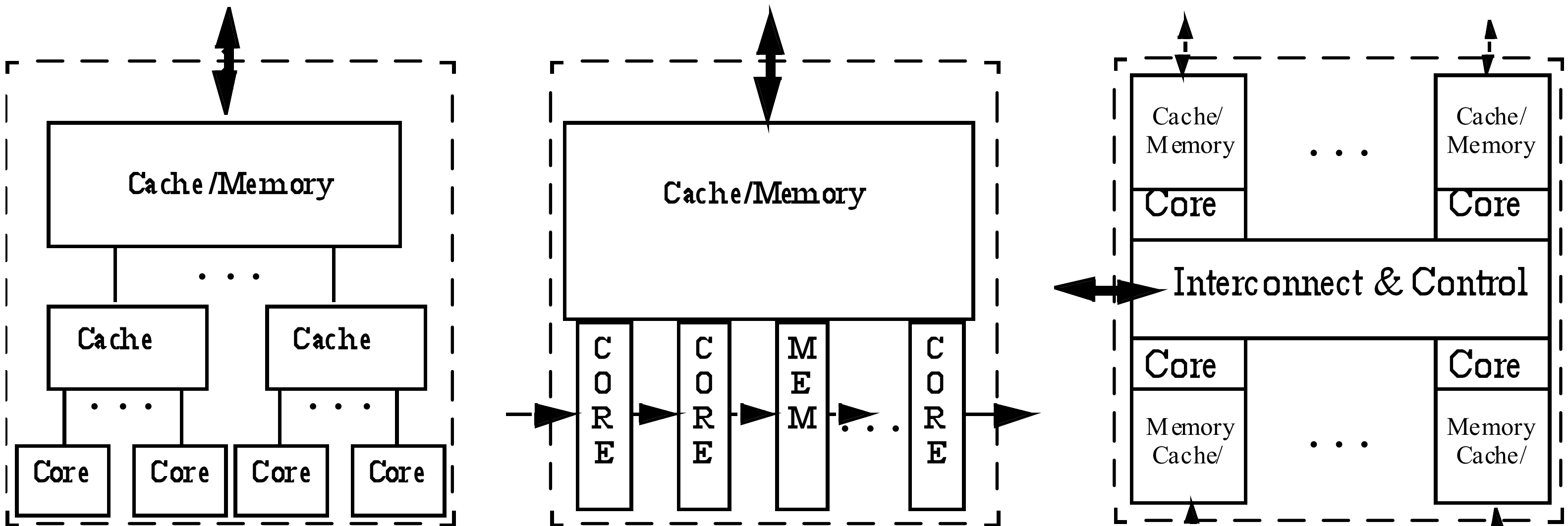


UNIVERSITY OF NOTRE DAME

Loss of Clock Increase Left Brute Parallelism as Only Performance Option



The Multi-core Architectural Spectrum



(a) Hierarchical Designs

- Intel Core Duo
- IBM Power5
- AMD Opteron
- SUN Niagara
- …

(b) Pipelined Designs

- IBM Cell
- *Most Router chips*
- Many Video chips

(c) Array Designs

- Terasys
- *Execube*
- Yukon
- Intel Teraflop

And Then There's "Heterogeneous" Multi Core

UNIVERSITY OF NOTRE DAME

The DARPA Exascale Study

http://www.darpa.mil/tcto/docs/ExaScale_Study_Initial.pdf

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

The Exascale Study

Study Directive: can we ride silicon to Exa by 2015?

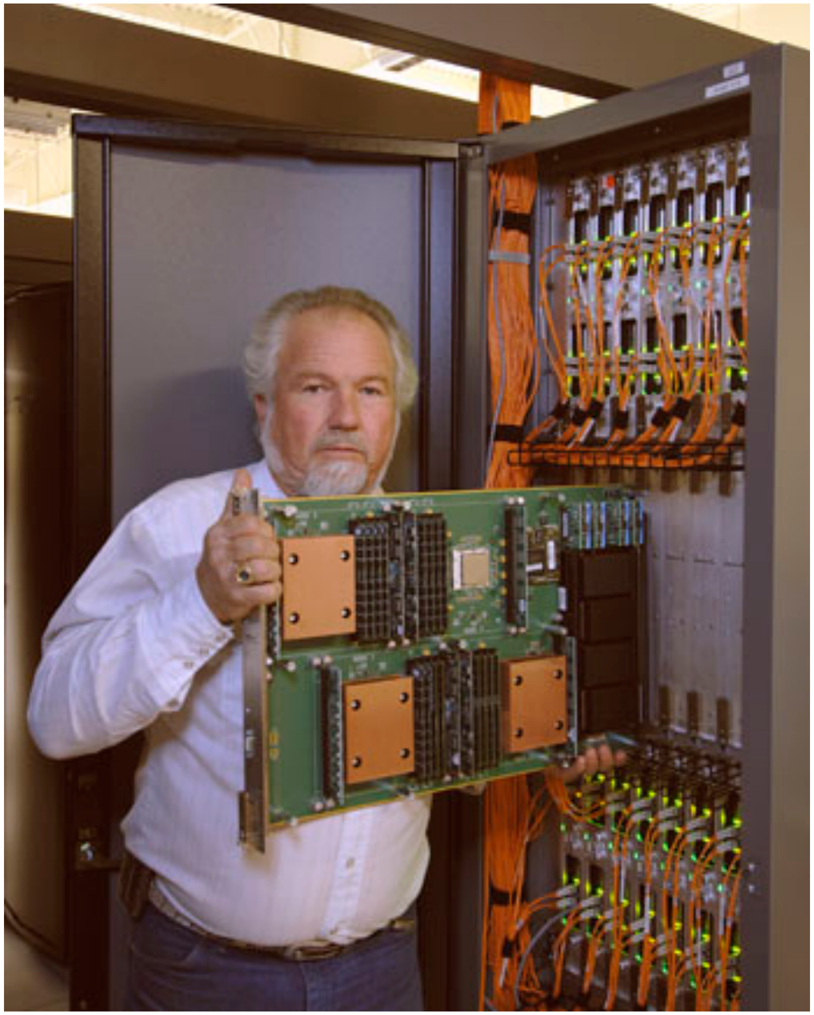
1. **Baseline today's:**
   - Commodity Technology
   - Architectures
   - Performance (Linpack)
2. **Articulate scaling of potential application classes**
3. **Extrapolate roadmaps for**
   - "Mainstream" technologies
   - Possible offshoots of mainstream technologies
   - Alternative and emerging technologies
4. **Use technology roadmaps to extrapolate use in "strawman" designs**
5. **Analyze results & id "*Challenges*"**

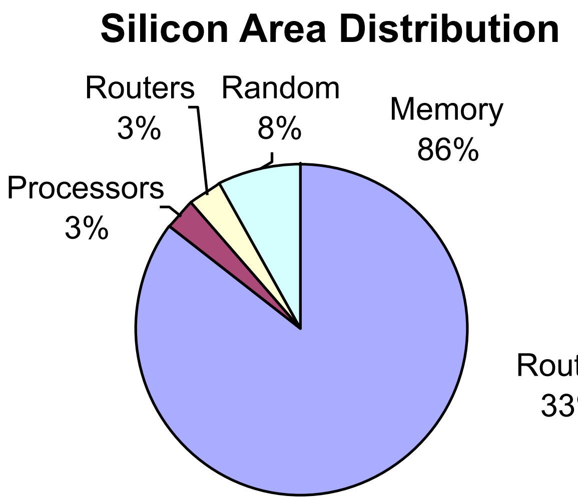
Exascale Strawmen

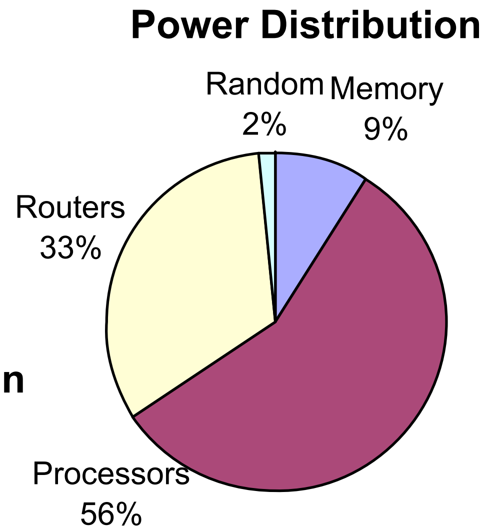
Architectures Considered

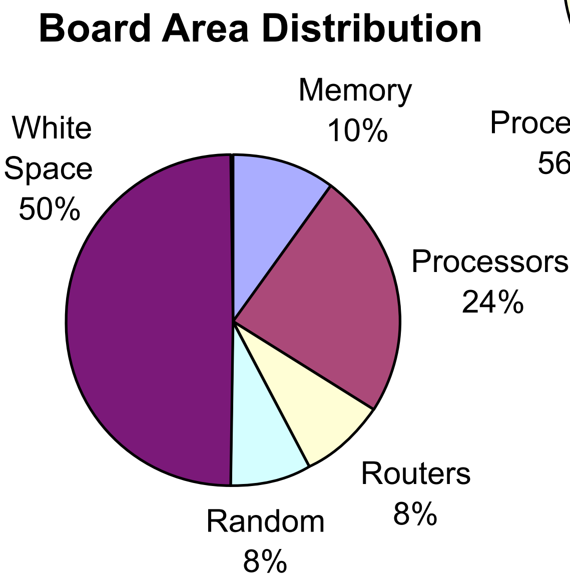
- Evolutionary Strawmen
  - "Heavyweight" Strawman based on commodity-derived microprocessors
  - "Lightweight" Strawman based on custom microprocessors
- Aggressive Strawman
  - "Clean Sheet of Paper" CMOS Silicon

A Modern "Heavyweight" HPC System



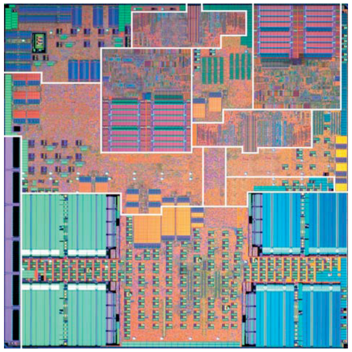






A "Light Weight" Node Alternative

2 Nodes per "Compute Card." Each node:
- A low power compute chip
- Some memory chips
- "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

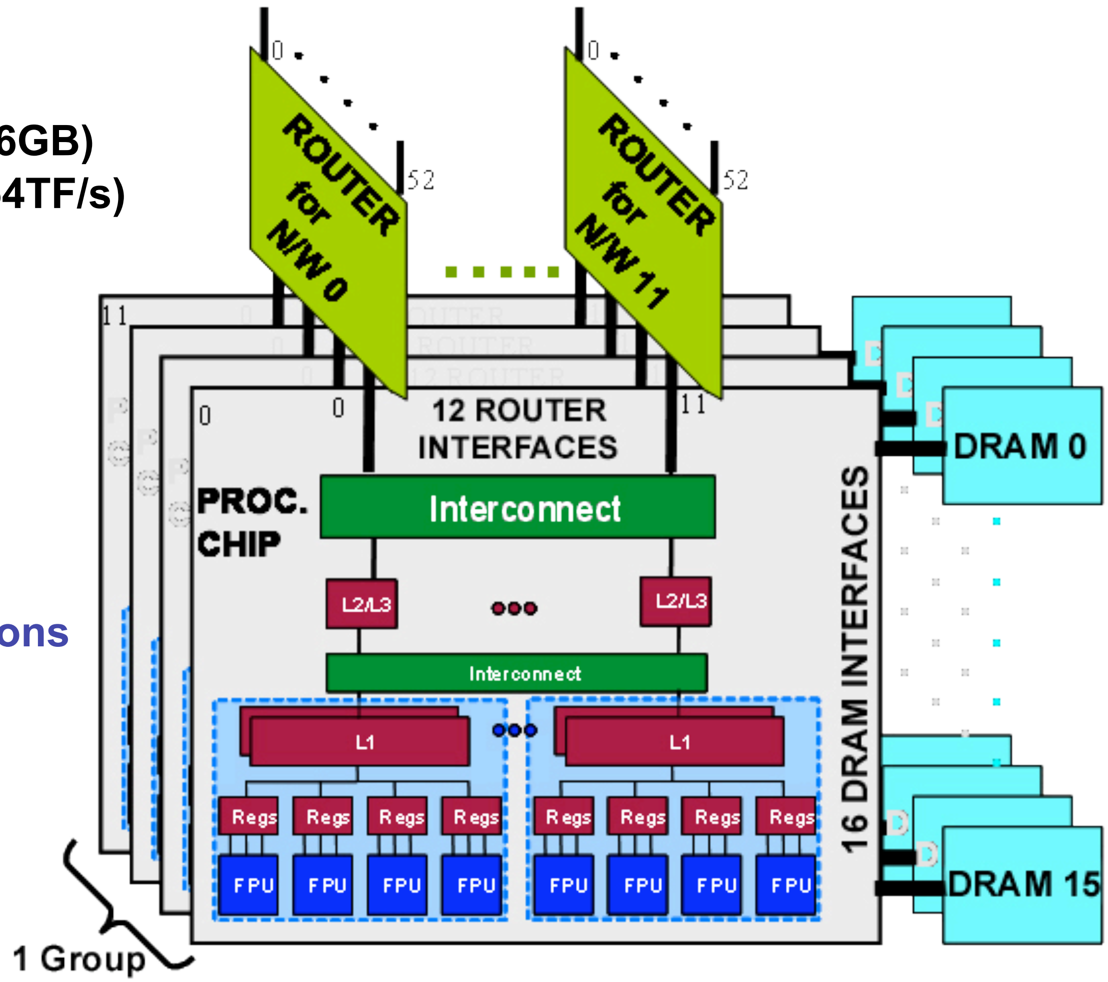
System Architecture:
- Multiple Identical Boards/Rack
- Each board holds multiple Compute Cards
- "Nothing Else"

"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

1 EFlop/s "Clean Sheet of Paper" Strawman

Sizing done by "balancing" power budgets with achievable capabilities

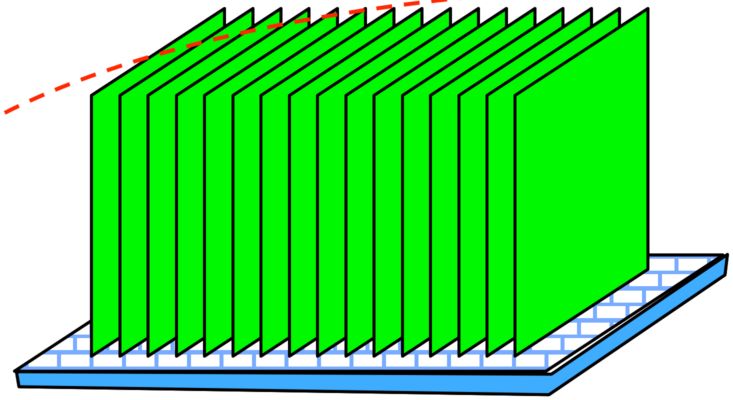
- 4 FPUs+RegFiles/Core (=6 GF @1.5GHz)
- 1 Chip = *742* Cores (=4.5TF/s)
  - 213MB of L1I&D; 93MB of L2
- 1 Node = 1 Proc Chip + 16 DRAMs (16GB)
- 1 Group = 12 Nodes + 12 Routers (=54TF/s)
- 1 Rack = 32 Groups (=1.7 PF/s)
  - 384 nodes / rack
- 3.6EB of Disk Storage included
- 1 System = 583 Racks (=1 EF/s)
  - 166 MILLION cores
  - 680 MILLION FPUs
  - 3.6PB = 0.0036 bytes/flops
  - 68 MW w'aggressive assumptions



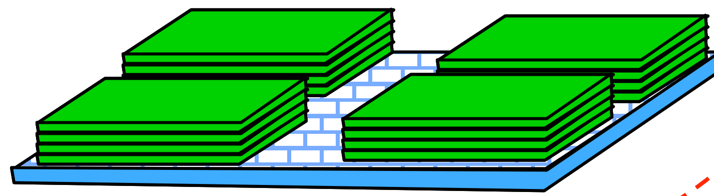
Largely due to Bill Dally, Stanford

A Single Node (No Router)

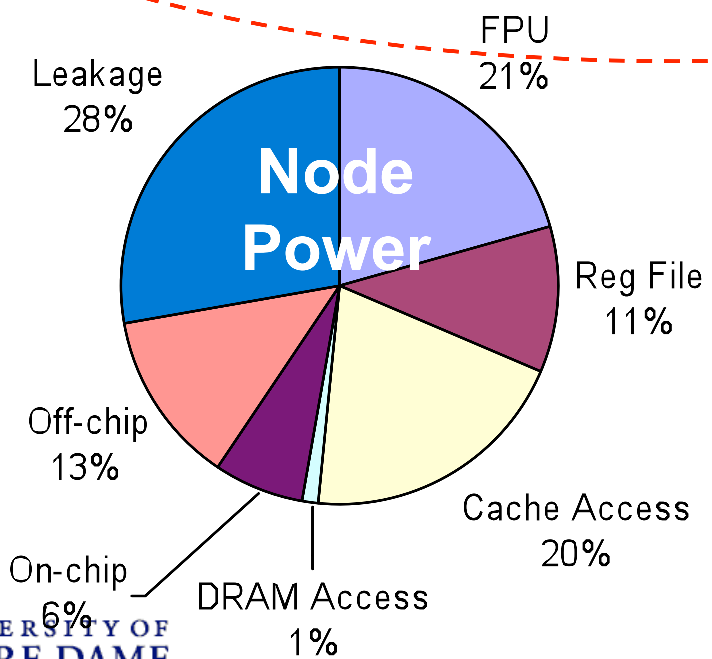
“Stacked” Memory



(a) Quilt Packaging



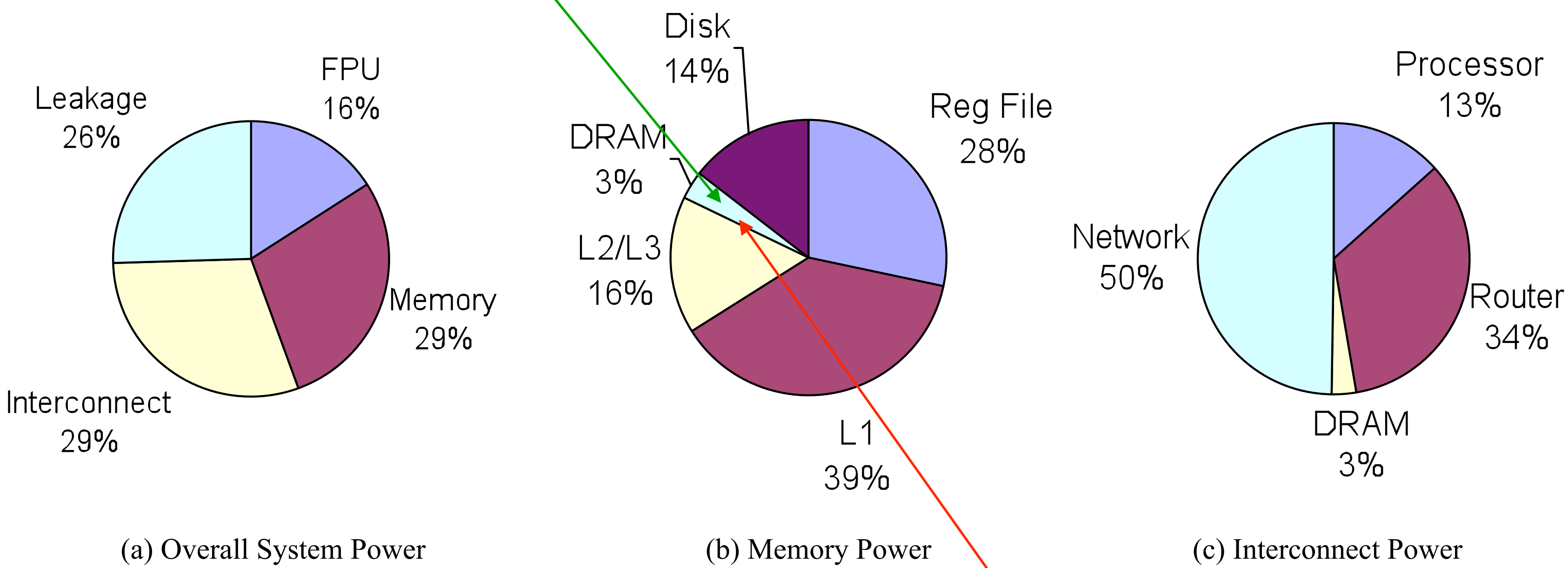
(b) Thru via chip stack



Characteristics:
- **742 Cores; 4 FPUs/core**
- **16 GB DRAM**
- **290 Watts**
- **1.08 TF Peak @ 1.5GHz**
- **~3000 Flops per cycle**

Strawman Power Distribution

Assumed a 60X per bit decrease in access energy

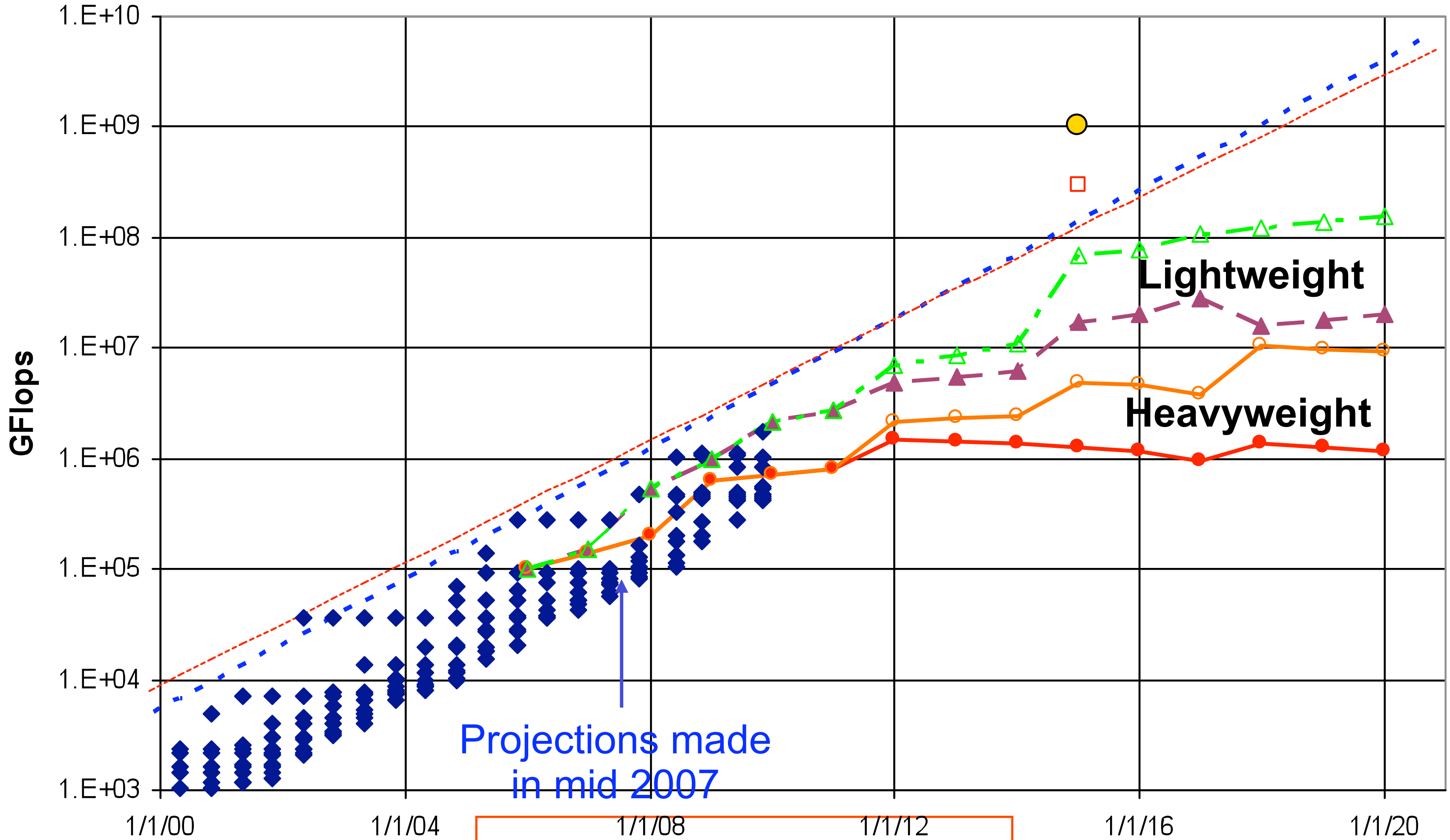


(a) Overall System Power (b) Memory Power (c) Interconnect Power

- 12 nodes per group
- 32 groups per rack
- 583 racks
- 1 EFlops/3.6 PB
- ***166 million cores***
- ***67 MWatts***

Growing DRAM capacity 30X at least *doubles* overall Memory Power

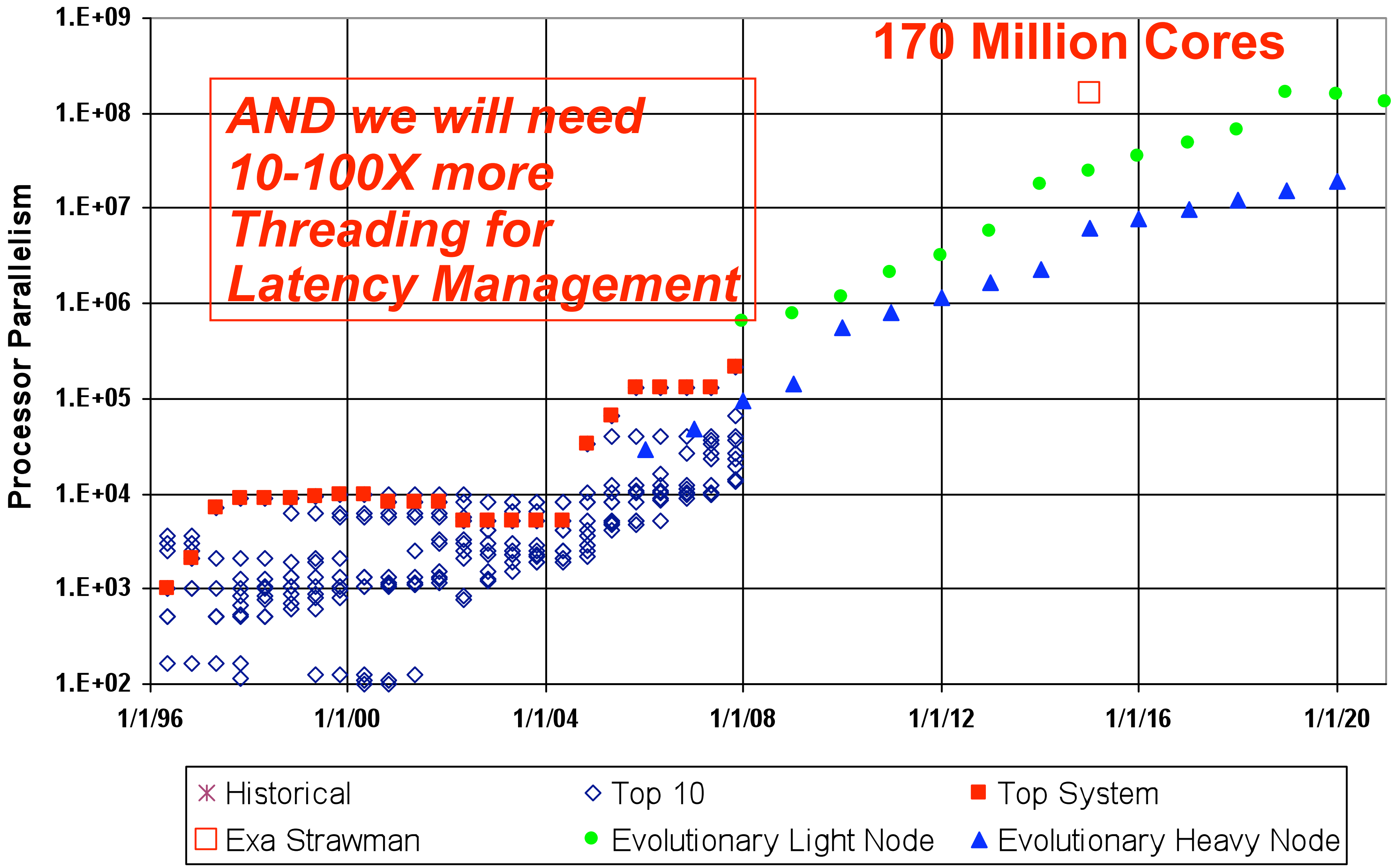
Data Center Performance Projections



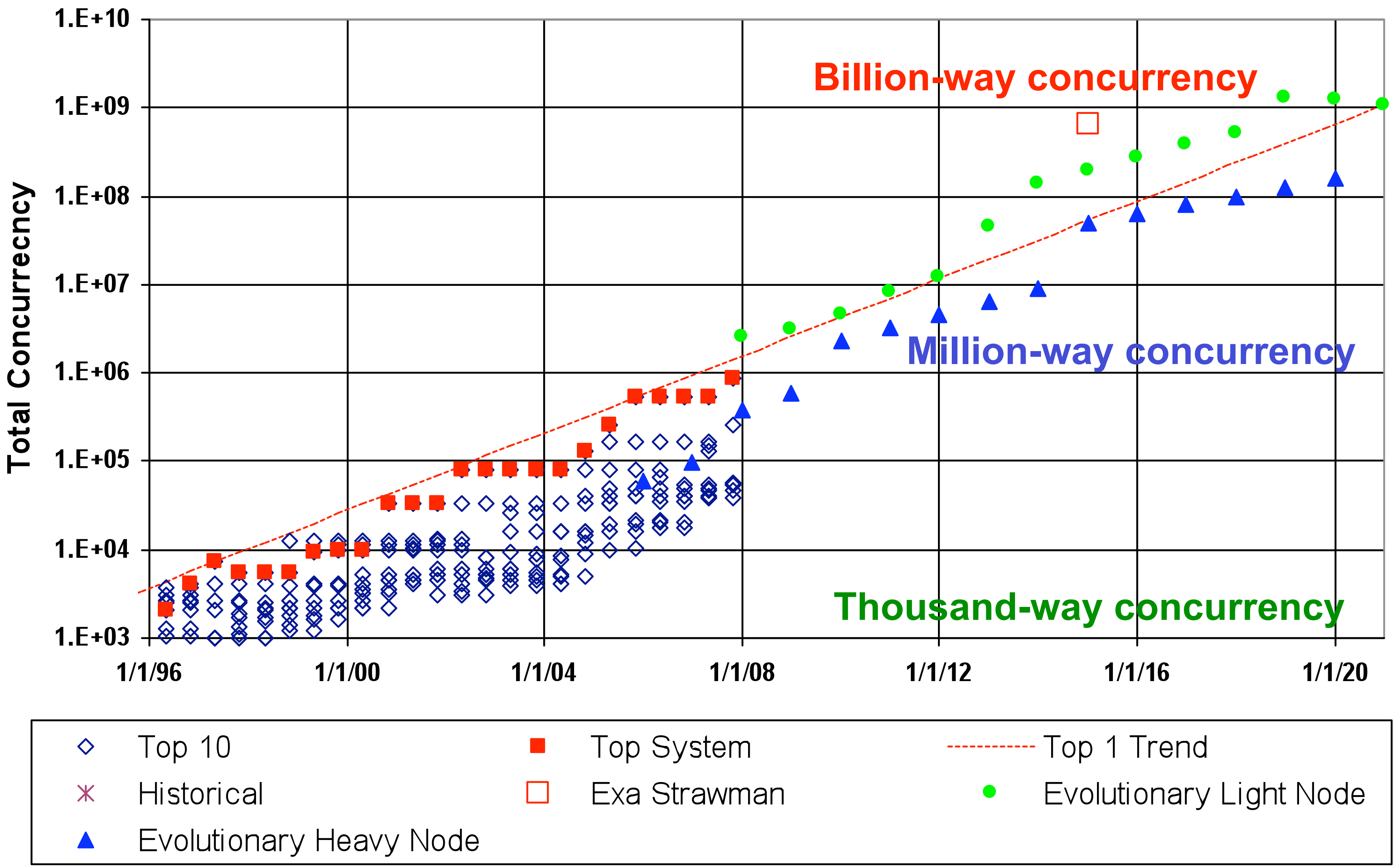
But not at 20 MW!

UNIVERSITY OF NOTRE DAME

Data Center Core Parallelism

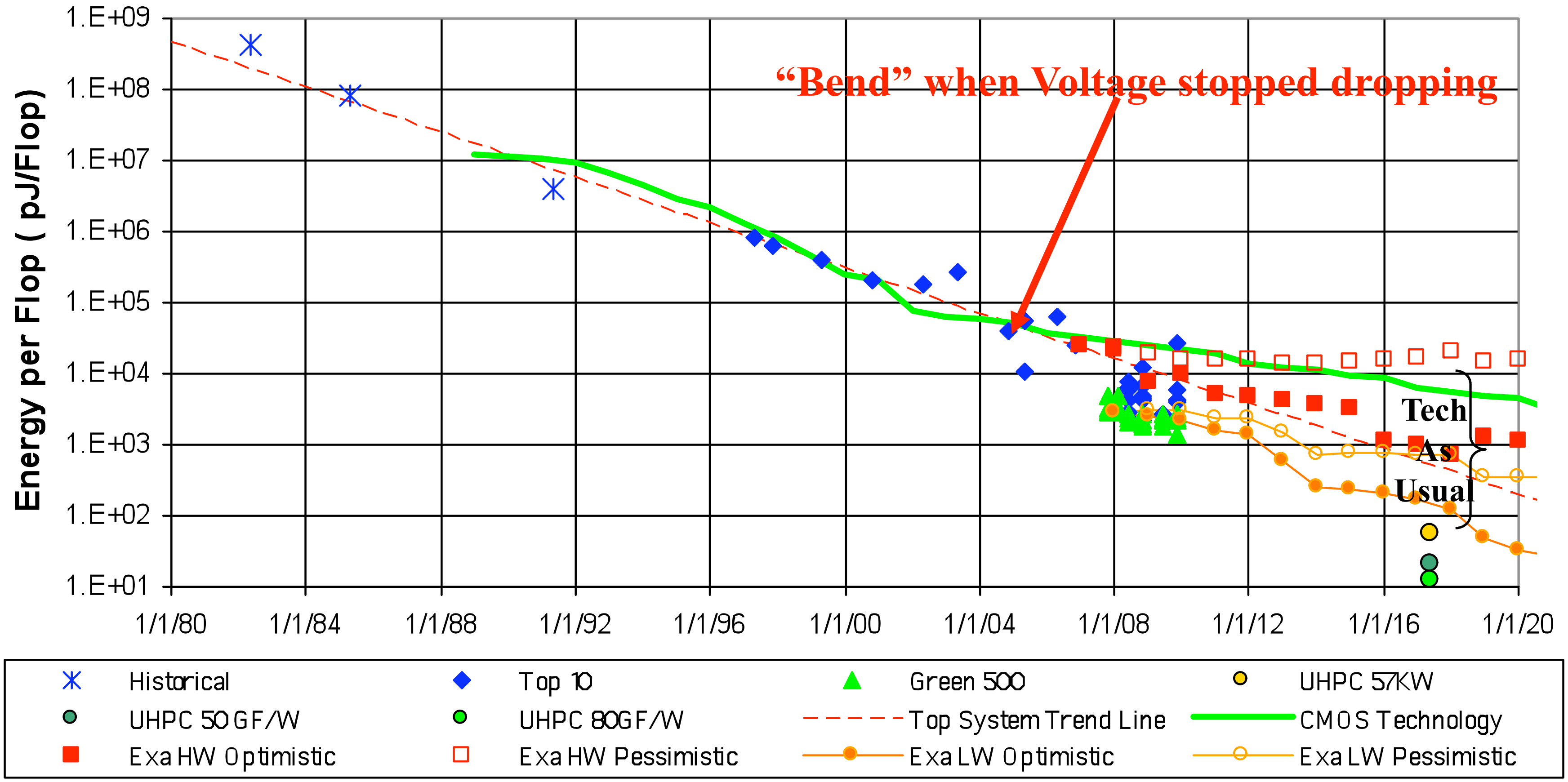


Data Center Total Concurrency

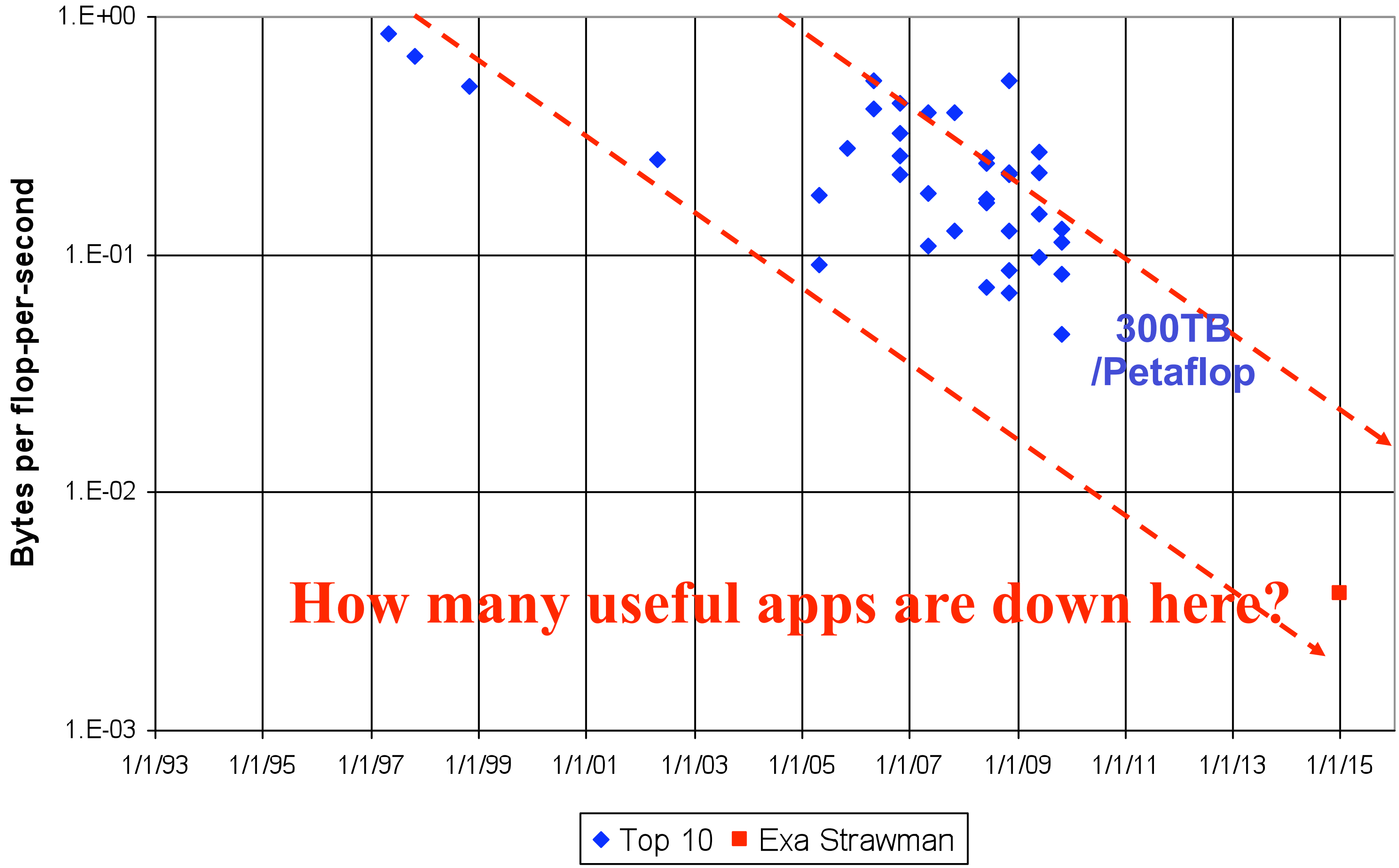


UNIVERSITY OF NOTRE DAME

“Business as Usual” Energy Projections



But At What Memory Capacity?



Key Take-Aways

- **Developing Exascale systems will be really tough**
  - In any time frame, for any of the 3 classes
- **Evolutionary Progression is at best 2020ish**
  - With limited memory
- **4 key challenge areas**
  - ***<u>Power!!!!!!!!</u>***
  - ***Concurrency***
  - ***Memory Capacity***
  - ***Resiliency***
- **Will require coordinated, cross-disciplinary ~~efforts~~**

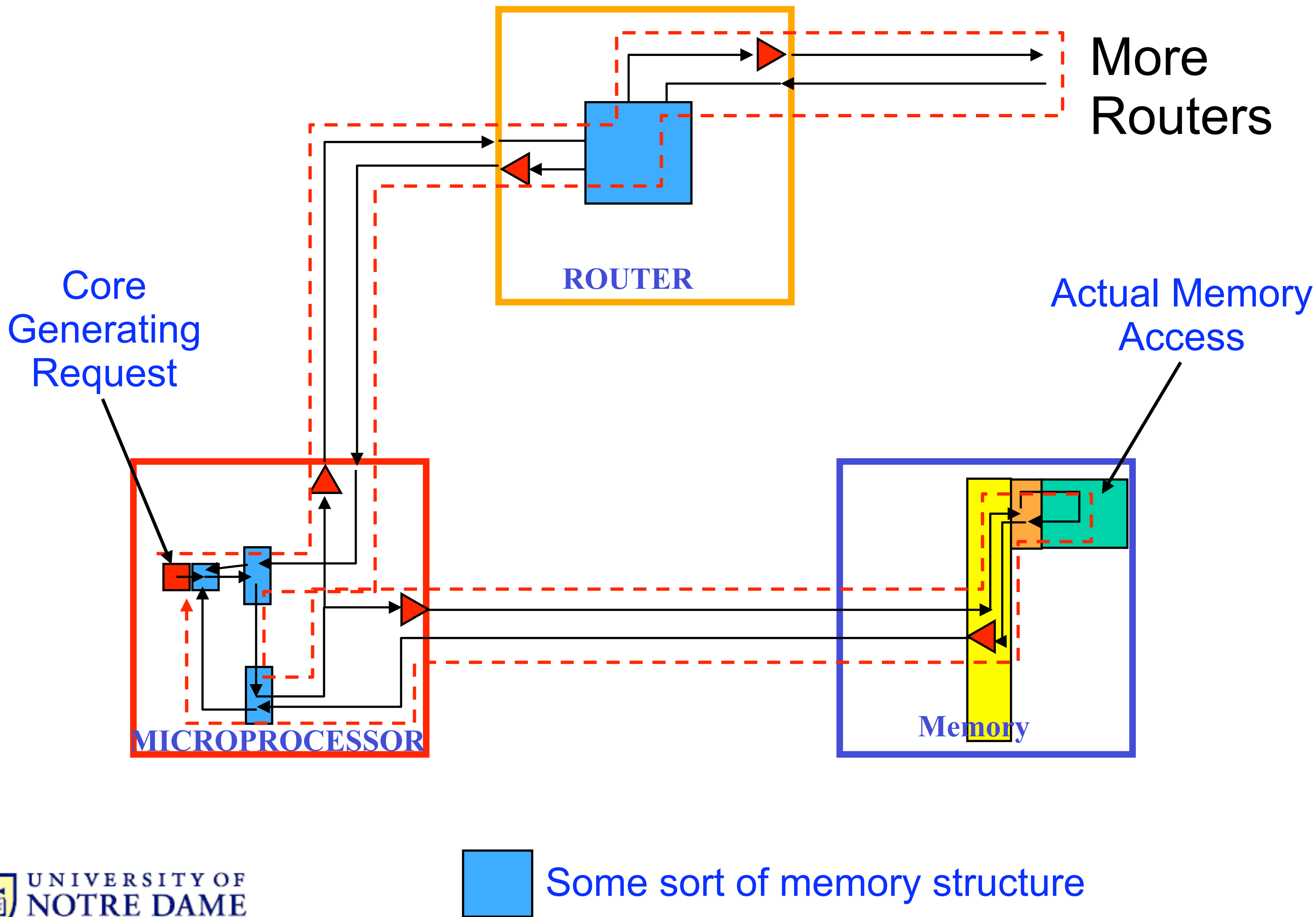
MIRACLES

A Deep Dive into Interconnect to Deliver Operands

Assumed Technology Numbers

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

Aggressive Strawman Access Path: Interconnect-Driven



Tracking the Energy

0	Total (pJ)	RF Access	Router Logic	Tag Access	32KB SRAM Access	DRAM Access	On-Chip Transport	TSV	Chip-Board-Chip	Chip-Optical-Chip
L1 Hit	39	30%	0%	28%	42%	0%	0%	0%	0%	0%
L2/L3 Hit	385	7%	0%	6%	34%	0%	53%	0%	0%	0%
Local	1380	1%	0%	2%	2%	68%	26%	1%	0%	0%
Global	13819	0%	24%	0%	0%	1%	13%	0%	10%	52%
Ave Read	706	2%	19%	2%	4%	8%	16%	0%	8%	41%
Write to L1	39	30%	0%	28%	42%	0%	0%	0%	0%	0%
Flush L1	891	0%	0%	1%	26%	32%	39%	1%	0%	0%

Linpack Energy

- **Linpack: standard benchmark for TOP500 rankings**
- **Solve Ax=b by LU matrix decomposition & back solve**
- **Computationally intensive step: update rows after pivot**
- **Basic function Y[j,k] = α Y[I,k] + Y[j,k]**
  - Read in C αs (C = “size” of cache line in dwords) (remote memory)
  - Read in C Y’s from row i (remote memory)
  - Repeat for all elements of each row k
    - Read in C Y’s from row k (local memory)
    - Do processing
    - Write C Y’s (to cache)
    - At some time flush C Y’s back to local memory from cache

Resulting Energy per Flop

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	121,400	36.3%
Read Other Y[i]s	L1	39	264	10,425	3.1%
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	

And a core flop in 2015 is only 10pJ!!!

The Rockster Algorithm

“Extreme computing” in another dimension

The Problem:
Take a Picture – Find the Rocks

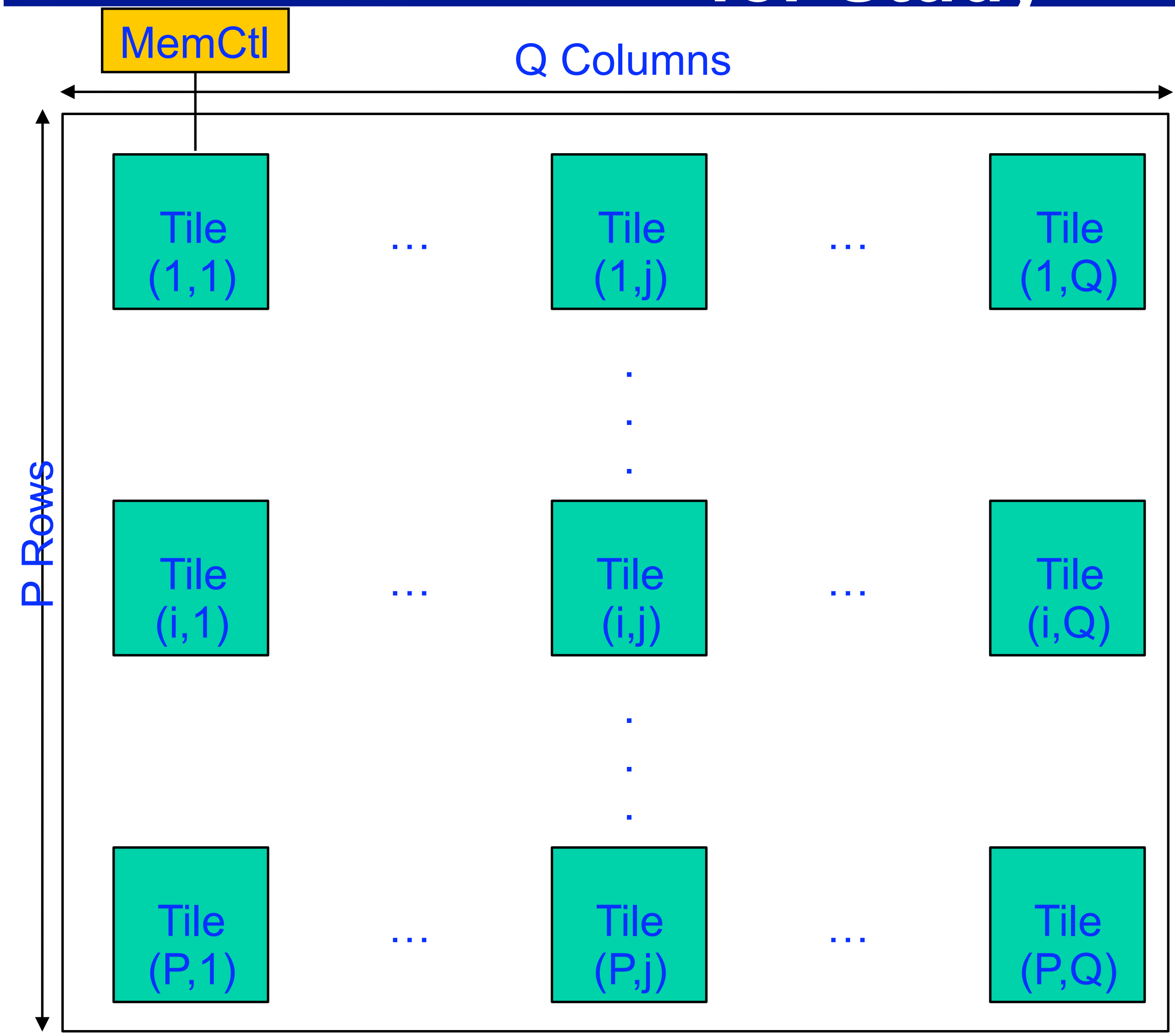
- NxN pixel Image (1Kx1K)
- Grey scale pixels (8bits)
- Output: “Rock Descriptions”
- Image starts in memory (streamed from sensors)
- Assume K rocks of E pixels each around edges
  - Ave rock area = (E/4)^2 pixels

The Rockster Algorithm

1. **Move the image from memory to the tiles**
2. **5x5 Gaussian smoothing**
3. **3x3 Canny edge sharpener**
    1. Include magnitude/angle computations per pixel
4. **Horizon determination (sky fill)**
    1. Homogeneous regions touching the ”top”
5. **Edge following, including gap fix**
6. **Background fill to generate binary “Rock Mask”**
    1. 1 bit/pixel: rock/not rock
    2. Write mask back to memory
7. **Connected Component to trace & report “Edge List” of rocks**

Algorithm from JPL

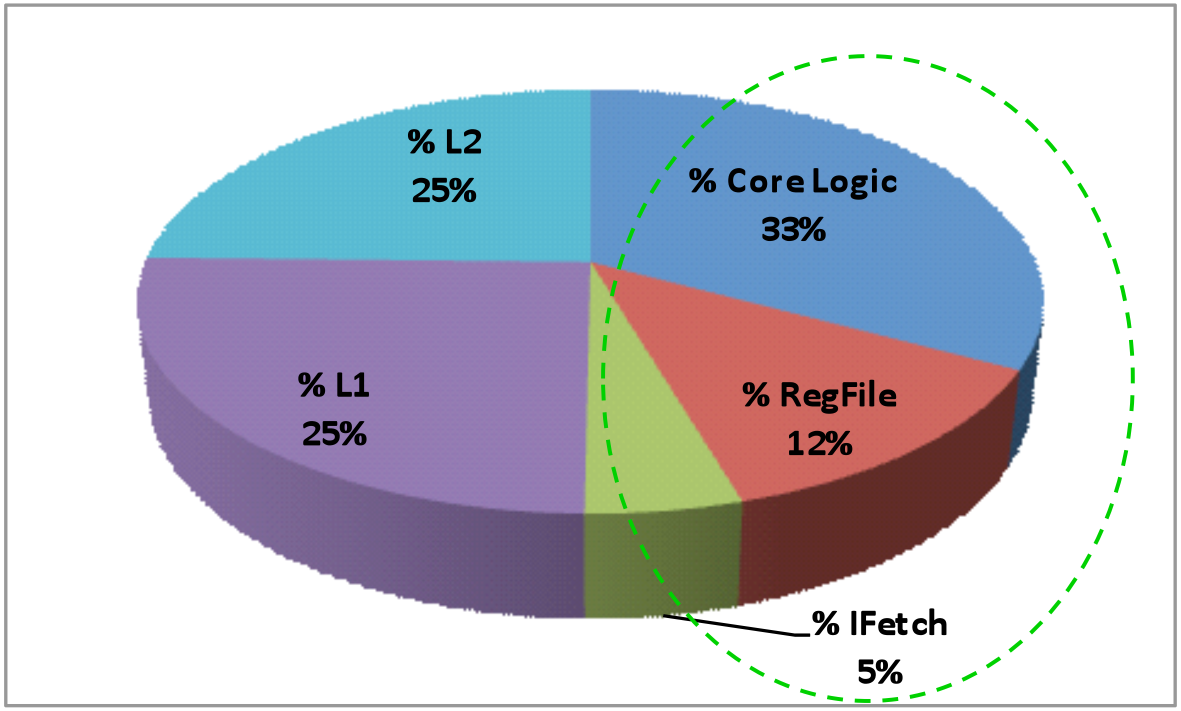
Simplified Tiled Multicore Architecture for Study



- PxQ "Tiles"
- All Tiles identical
    - Each 64KB L2 today
- Only X-Y neighbor routing
- Memory Access only from (1,1)

What Does Memory Access Energy Look Like ?

- Under some reasonable assumptions, with no off-tile memory references:



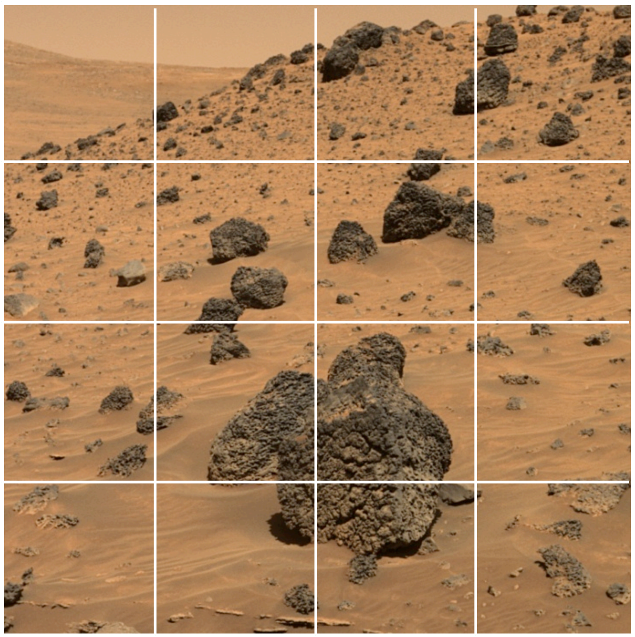
A "Compute Cycle" Of Energy

Each off-tile memory reference costs eqvt of an additional *212 compute cycles* of energy

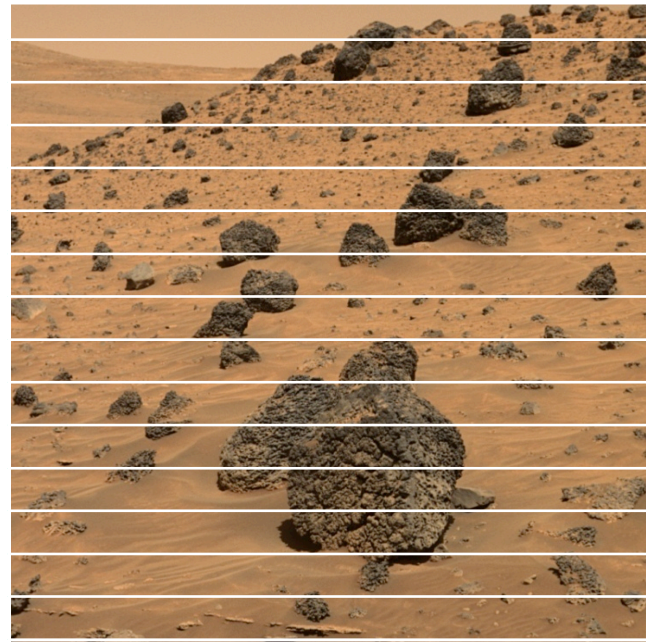
How Much Energy to Move the Image?

- **Assume PxP tiles**
- **Each tile accesses N^2/P^2 pixels = $2^{14}/P^2$ lines**
- **Each access:**
  - 1 read from DRAM + 1 write to L2:
  - Transport through (i + j -1) tiles:
- **Total: *5.4 + 0.15*P Million* compute cycles**
- **More tiles – more energy!**
  - Eg: P=10 (100 tiles): 25% energy adder

Partitioning the Image onto PxQ Tiles



(a) Subimage per Tile
$p = N/P; q = N/P$



(b) Slice per Tile
$p = N/PQ; q = N$

- Subimages of size "p" rows of pixels by "q" columns of pixels
  - Equivalently: $pxq = N^2/(PQ)$ pixels
- Partitioning works only if $2N^2KB/(PQ)$ < L2 cache size

Ghost Cells

- **Many steps need "neighboring cell values"**
- **Ghost cells: copies of neighboring cells from neighboring tiles**
- **Data exchanges costs energy**

Total	P				
Energy	2	4	8	16	32
Subimage	149,760	299,520	599,040	1,198,080	2,396,160
Sliced	78,336	92,160	147,456	368,640	1,253,376

Explosive growth in energy as we grow # of tiles!

Cache Access Energy for Filters

Tiles	4	16	64	256	1024
Filter	P				
Energy	2	4	8	16	32
Subimage	3.9E+07	3.9E+07	3.9E+07	4.0E+07	4.0E+07
Sliced	3.2E+07	3.2E+07	3.2E+07	3.2E+07	3.2E+07

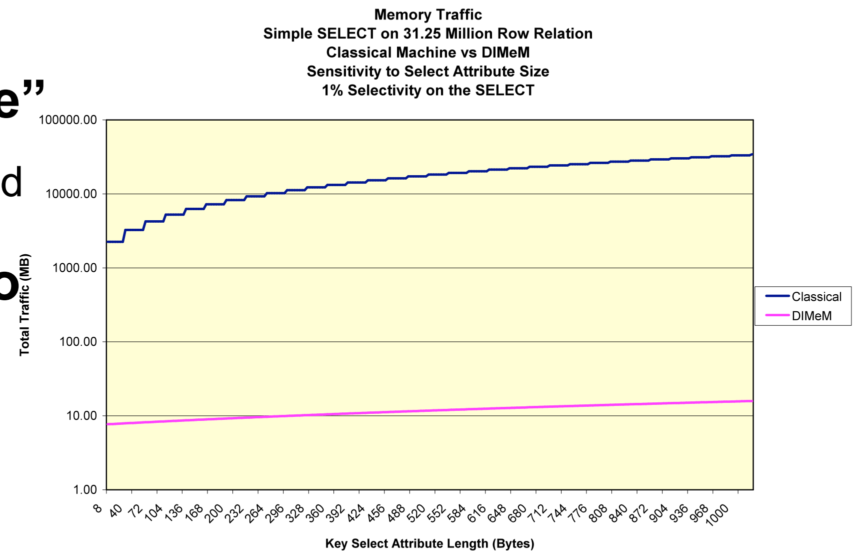
- **Above #s in units of equivalent compute cycles, summed over all tiles**
- **Equivalent to about 30-40 compute cycles of energy *per pixel***
- **Remember need to add in Ghost energy**

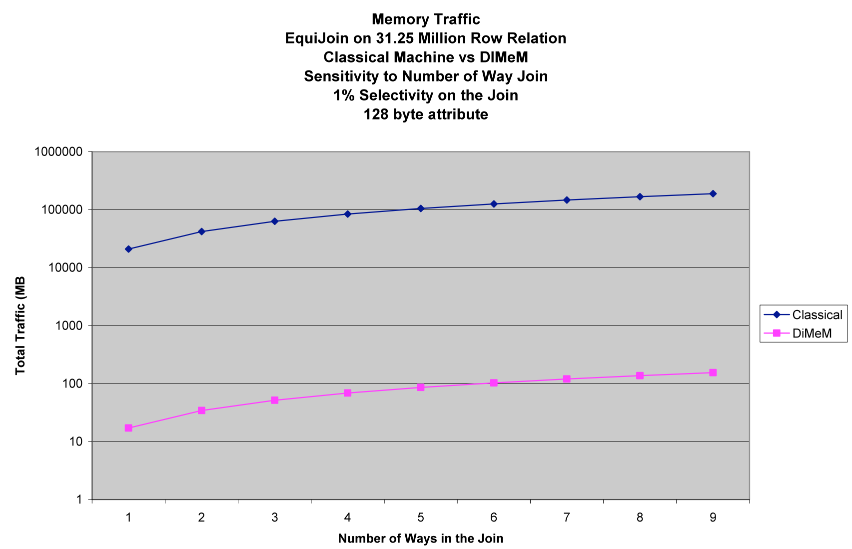
Rockster Conclusions

- **Computation is cheap energy-wise**
- **Accessing local memory is expensive**
- **Accessing DRAM global memory is huge!**
- **The way you partition data affects energy**
- **Some odd, counter-intuitive effects**
  - Some energy grows as # tiles increase

Are There Alternative Architectures? Esp. for SQL-Like Operations

- **Distribute data through large "Partitioned Global Address Space"**
  - Each pGAS node has chunk of memory and some "anonymous" local processing
- **Use MAP-REDUCE like algorithm to distribute query**
  - On basis of address
- **Keep results local until requested**
- **Common result:** ***orders of magnitude reduction in link traffic***





DON'T MOVE THE DATA!

New DARPA Program

- **UHPC - Ubiquitous High Performance Computing**
- **Focii on "Extreme Scale" (1000X) technology**
  - power consumption
  - cyber resiliency
  - productivity
- **Target research machines by 2018**
  - Petascale in a rack
  - Terascale module
- **Challenge problems**
  - streaming sensor data
  - large dynamic graph-based informatics problem
  - decision class problem that encompasses search, hypothesis testing, and planning
  - 2 problems from HPCMP or CREATE suites
- **Award announcements very shortly**

Overall Conclusions

- **Exascale needed for Astroinformatics**
- **World has gone to multi-core to continue Moore's Law**
- **Pushing performance another 1000X will be tough**
- **The major problem is in energy**
- **And that energy is in *memory & interconnect***
  - For both scientific and embedded applications
  - And even for data search applications
- **We need to begin rearchitecting to reflect this**
  - Need alternative metrics to "flops"
  - ***DON'T MOVE THE DATA!***
- **We need to begin redesigning our architectures & *codes* for energy minimization – now!**
  - Map-Reduce a promising paradigm