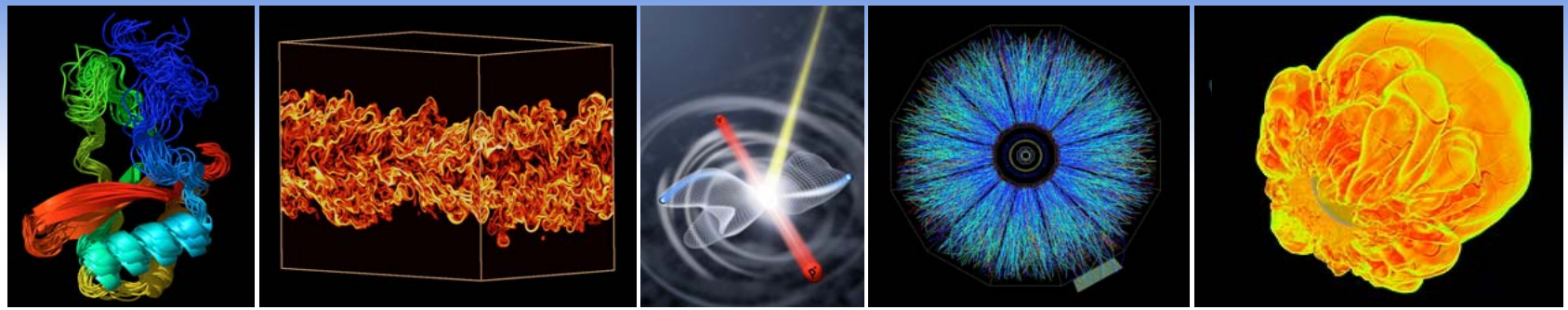


Multicore in HPC: Opportunities and Challenges

Katherine Yelick
NERSC Director
Lawrence Berkeley National Laboratory

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC

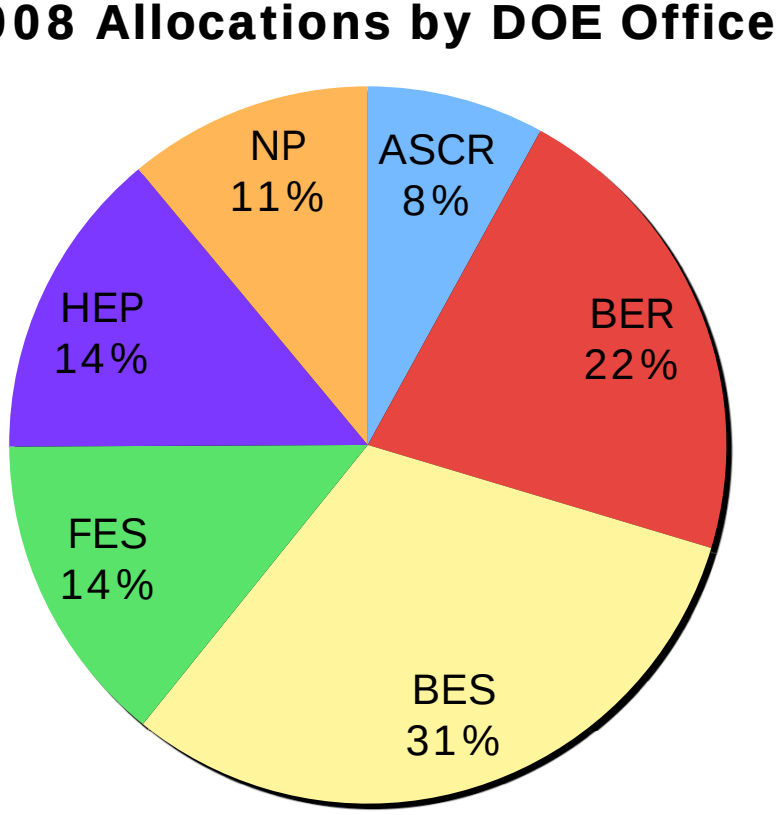
BERKELEY LAB

NERSC Mission

NERSC was established in 1974. Its mission is to *accelerate the pace of scientific discovery* by providing high performance computing, information, data, and communications services for *all DOE Office of Science (SC) research.*

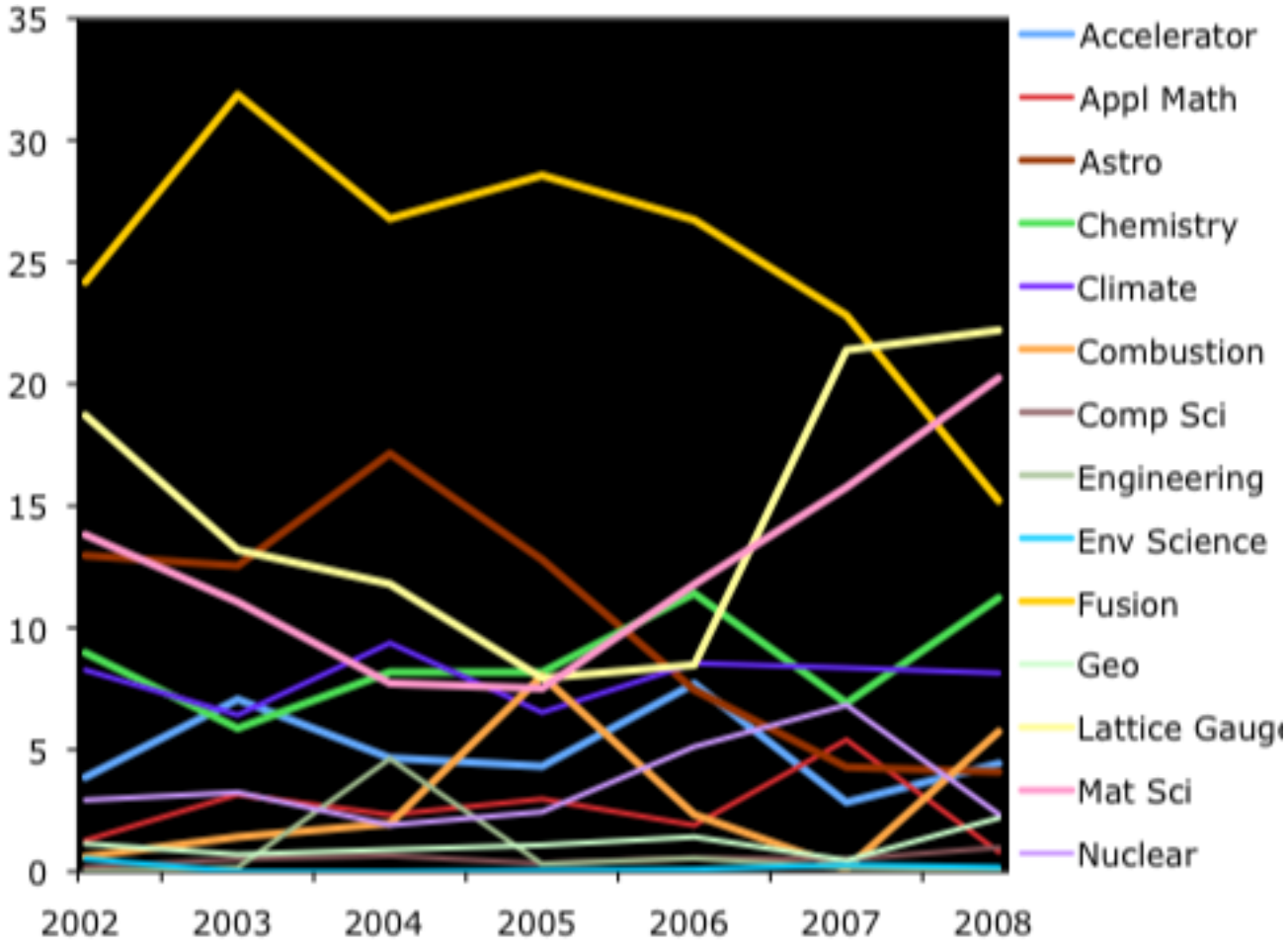
NERSC is the Production Facility for DOE Office of Science

- NERSC serves a large population of users
  - ~3000 users, ~400 projects, ~500 codes
- Allocations by DOE
  - 10% INCITE awards:
    - Created at NERSC
    - Open to all of science, not just DOE
    - Large allocations, extra service
  - 70% Production (ERCAP) awards:
    - From 10K hour (startup) to 5M hour
  - 10% each NERSC and DOE reserve
- Award mixture offers
  - High impact through large awards
  - Broad impact across domains



NERSC Serves Broad and Varying DOE Science Priorities

Usage by Science Area as a Percent of Total Usage



NERSC 2008 Configuration

Large-Scale Computing System

Franklin (NERSC-5): Cray XT4

Upgraded from Dual to Quad Core

- 9,740 nodes; 38,760 cores
- 9,660 computational nodes (38,640 cores)
- 79 TBs Aggregate Memory (8 GB per node)
- ~38 Tflops/s sustained SSP (355 Tflops/s peak)

Clusters

Bassi (NCSb)

- IBM Power5 (888 cores)

Jacquard (NCSa)

- LNXI Opteron (712 cores)

PDSF (HEP/NP)

- Linux cluster (~1K cores)

NERSC Global Filesystem (NGF)

- 230 TB; 5.5 GB/s
- IBM's GPFS

HPSS Archival Storage

- 74 PB capacity
- 11 Sun robots
- 130 TB disk cache

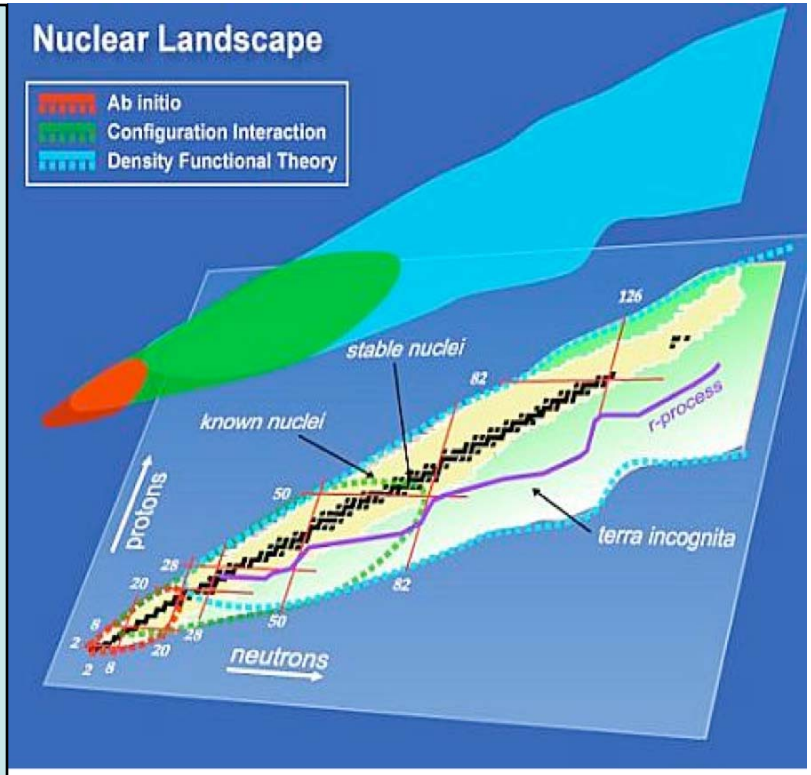
Analytics & Visualization

- Davinci (SGI Altix)

Nuclear Physics

- Calculation: High accuracy *ab initio* calculations on O^{16} using no -core shell model and no-core full configuration interaction model
- PI: James Vary, Iowa State

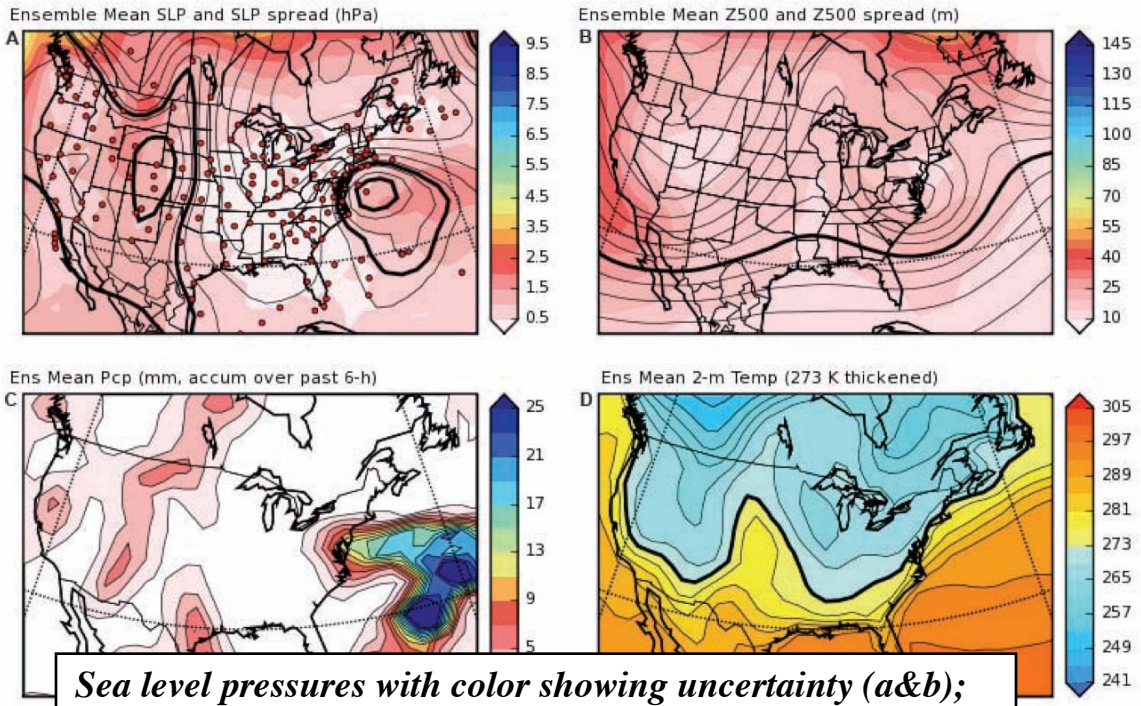
- **Science Results:**
  - Most accurate calculations to date on this size nuclei
  - Can be used to parametrize new density functionals for nuclear structure simulations
- **Scaling Results:**
  - 4M hours used in 2007
  - 12K cores; vs 2-4K before Franklin uncharged time
  - Diagonalize matrices of dimension up to 1 billion



Validating Climate Models

- INCITE Award for “20th Century Reanalysis” using an Ensemble Kalman filter to fill in missing climate data since 1892
- PI: G. Compo, U. Boulder

- **Science Results:**
  - **Reproduced 1922 Knickerbocker storm**
  - **Data can be used to validate climate and weather models**
- **Scaling Results:**
  - **3.1M CPU Hours in allocation**
  - **Scales to 2.4K cores**
  - **Switched to higher resolution algorithm with Franklin access**



Sea level pressures with color showing uncertainty (a&b); precipitation (c); temperature (d). Dots indicate measurements locations (a).

Low-Swirl Burner Simulation

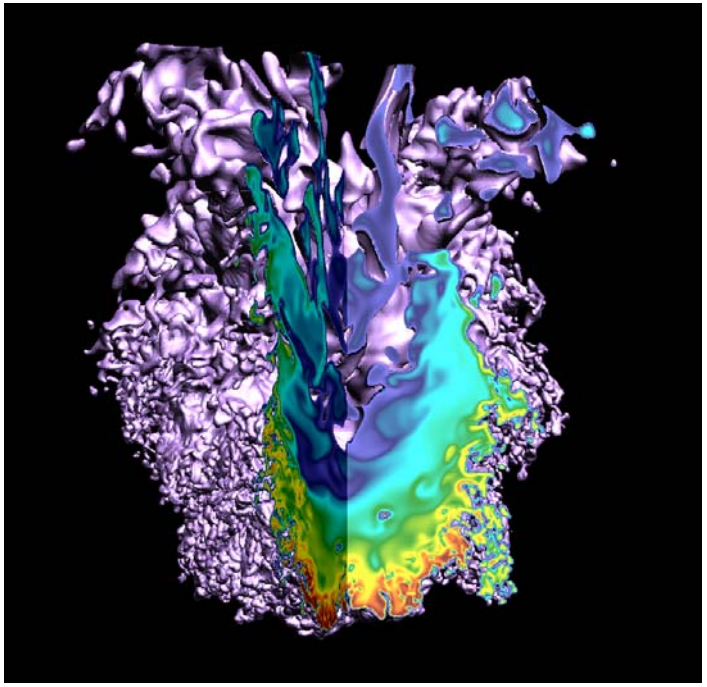
- Numerical simulation of a lean premixed hydrogen flame in a laboratory-scale low-swirl burner (LMC code)
- Low Mach number formulation with adaptive mesh refinement (AMR)
- Detailed chemistry and transport
- PI: John Bell, LBNL

Science Result:

- **Simulations capture cellular structure of lean hydrogen flames and provide a quantitative characterization of enhanced local burning structure**

NERSC Results:

- **LMC dramatically reduces time and memory.**
- **Scales to 4K cores, typically run at 2K**
- **Used 2.2M hours on Franklin in 2007, allocated 3.4M hours in 2008**

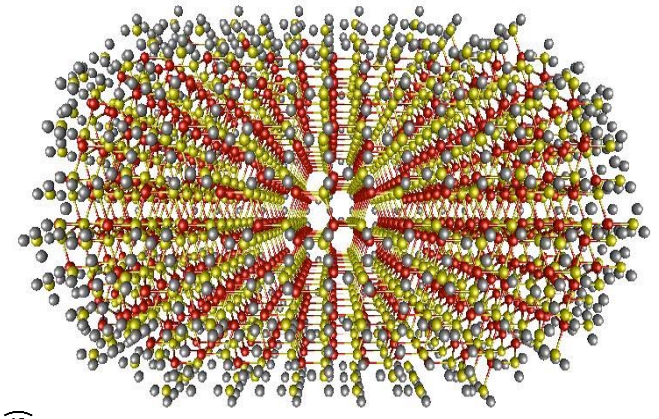


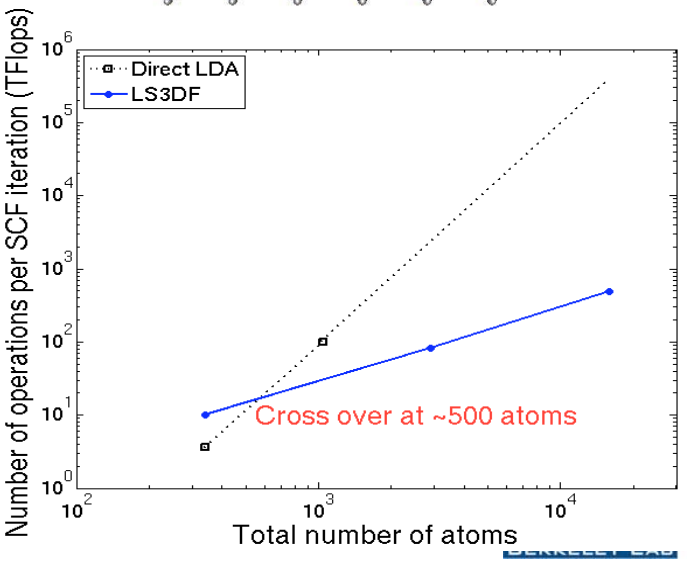
J B Bell, R K Cheng, M S Day, V E Beckner and M J Lijewski, Journal of Physics: Conference Series 125 (2008) 012027

Nanoscience Calculations and Scalable Algorithms

- **Calculation: Linear Scaling 3D Fragment (LS3DF). Density Functional Theory (DFT) calculation numerically equivalent to more common algorithm, but scales with O(n) in number of atoms rather than $O(n^3)$**
- **PI: L.W. Wang, LBNL**

- **Science Results**
  - **Calculated dipole moment on 2633 atom CdSe quantum rod, $Cd_{961}Se_{724}H_{948}$.**
- **Scaling Results**
  - **Ran on 2560 cores**
  - **Took 30 hours vs many months for $O(n^3)$ algorithm**
  - **Good parallel efficiency (80% on 1024 relative to 64 procs)**

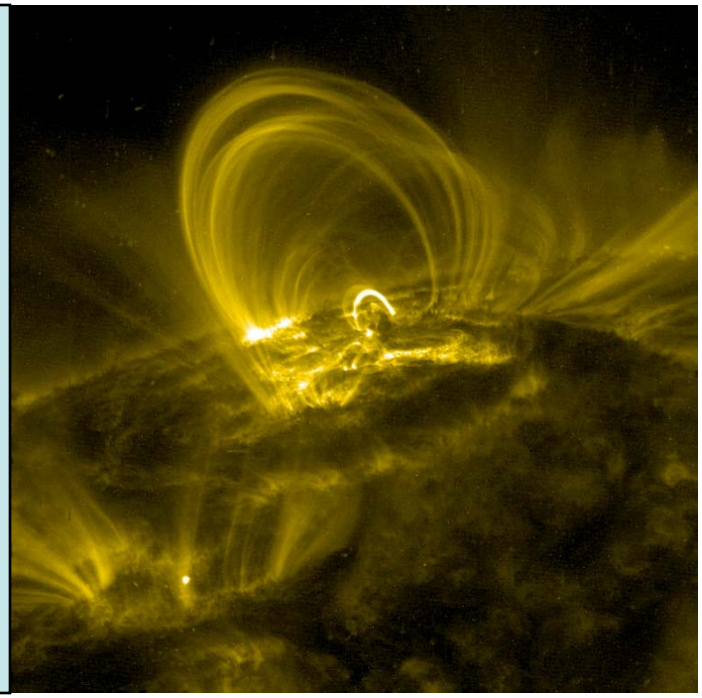




Astrophysics Simulation of Plasmas

- Calculations: AstroGK gyrokinetic code for astrophysical plasmas
- PIs: Dorland (U. of Maryland), Howes, Tatsuno

- **Science Results**
  - **Shows how magnetic turbulence leads to particle heating**
- **Scaling Results**
  - **Runs on 16K cores**
  - **Combines implicit and explicit methods**

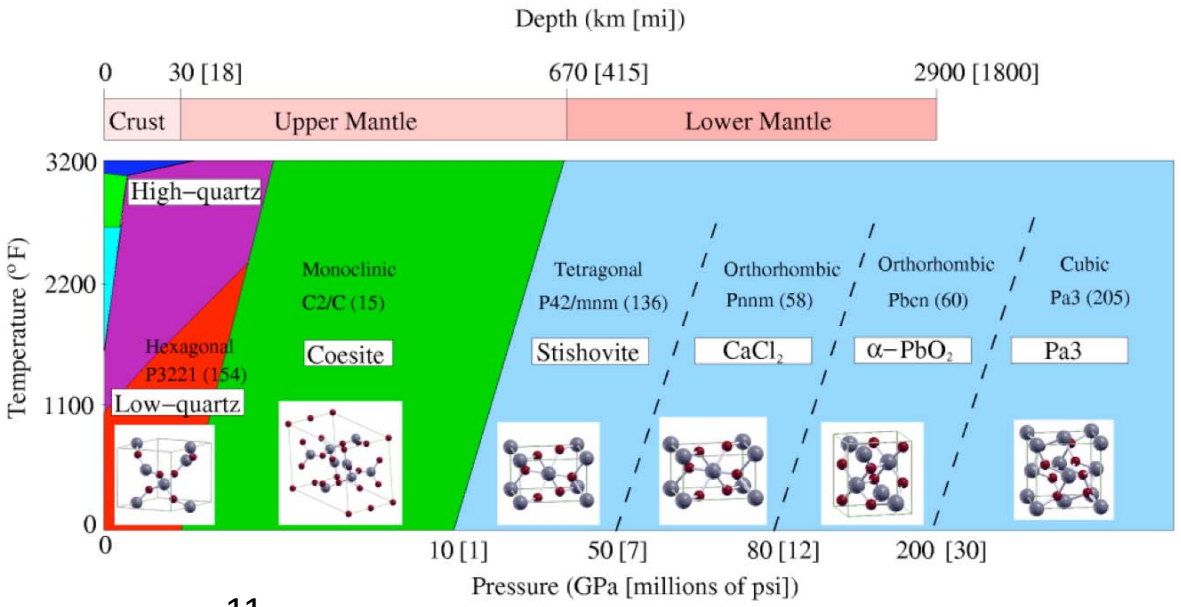


Modeling Dynamically and Spatially Complex Materials for Geoscience

- Calculation: Simulation of seismic waves through silicates, which make up 80% of the Earth's mantle
- PI: John Wilkins, Ohio State University

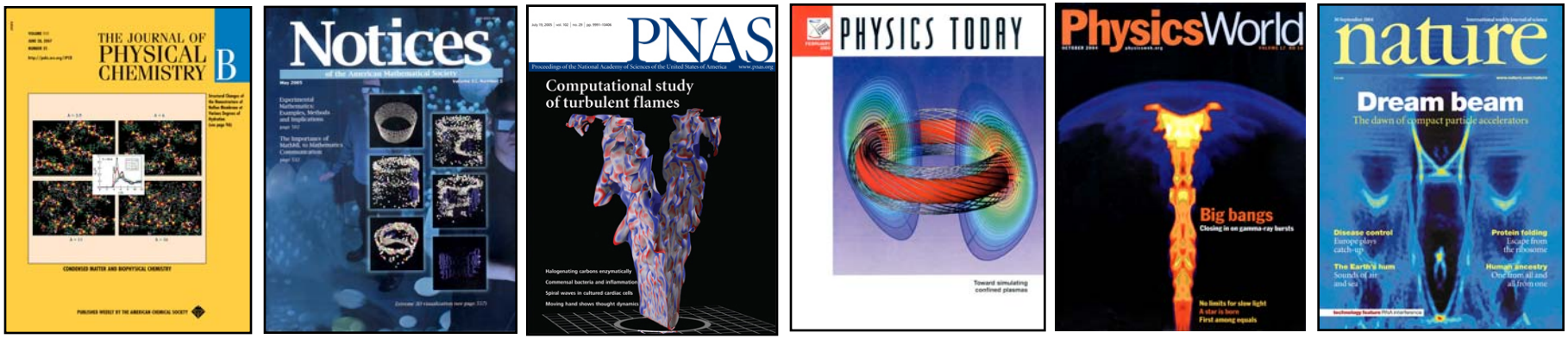
- **Science Result**
  - **Seismic analysis shows jumps in wave velocity due to structural changes in silicates under pressure**

- **Scaling Result**
  - **First use of Quantum Monte Carlo (QMC) for computing elastic constants**
  - **8K core parallel jobs in 2007**



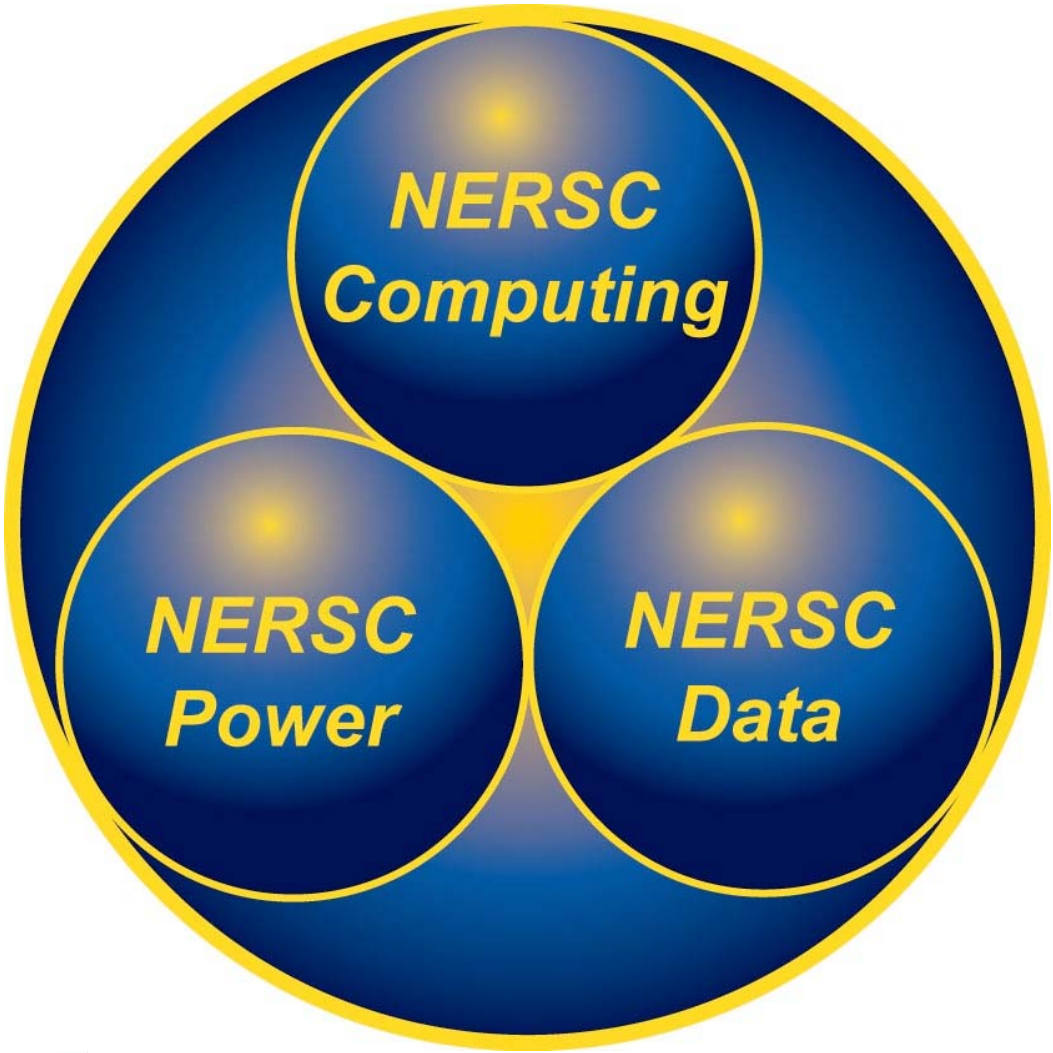
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Science Over the Years



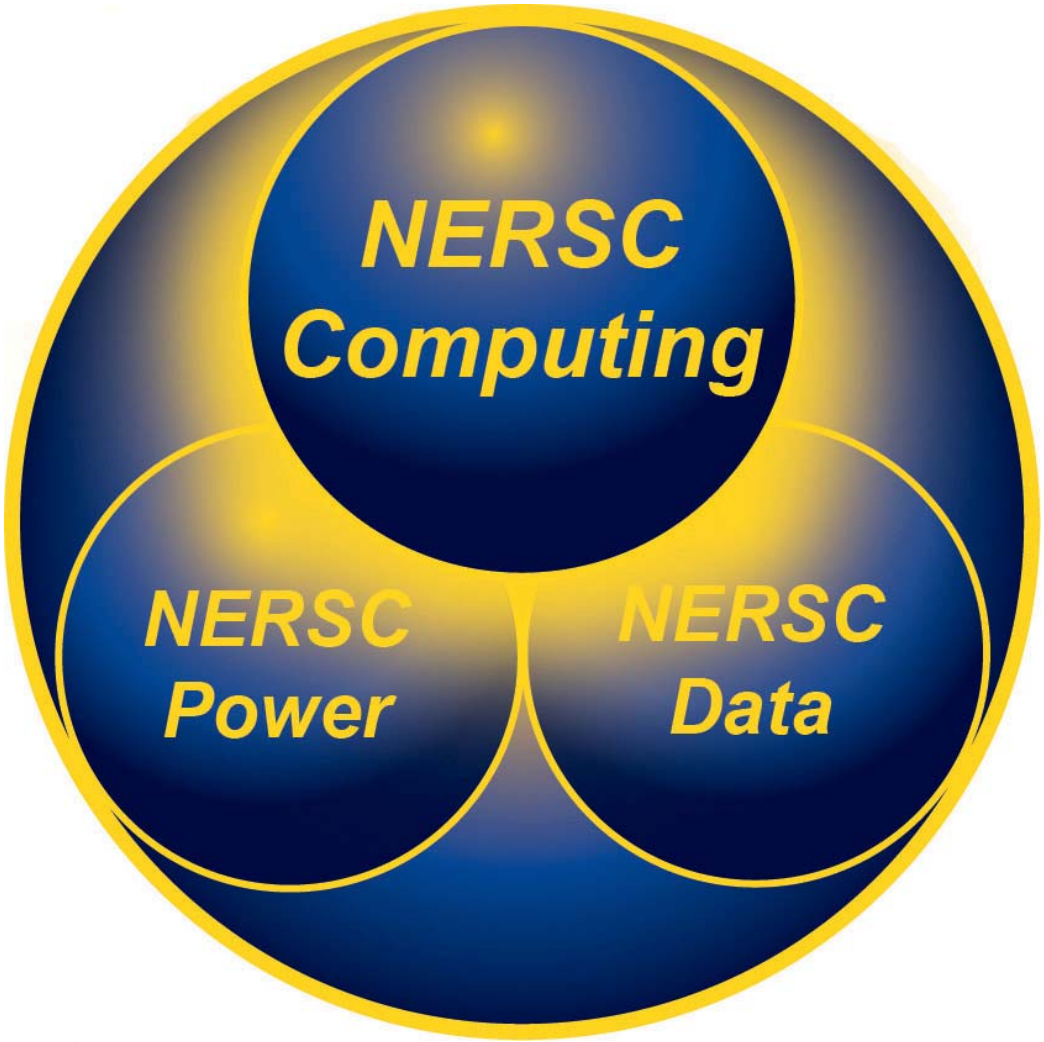
NERSC is enabling new science in all disciplines, with over *1,500 refereed publications* in 2007

NERSC Vision



NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

NERSC Computing

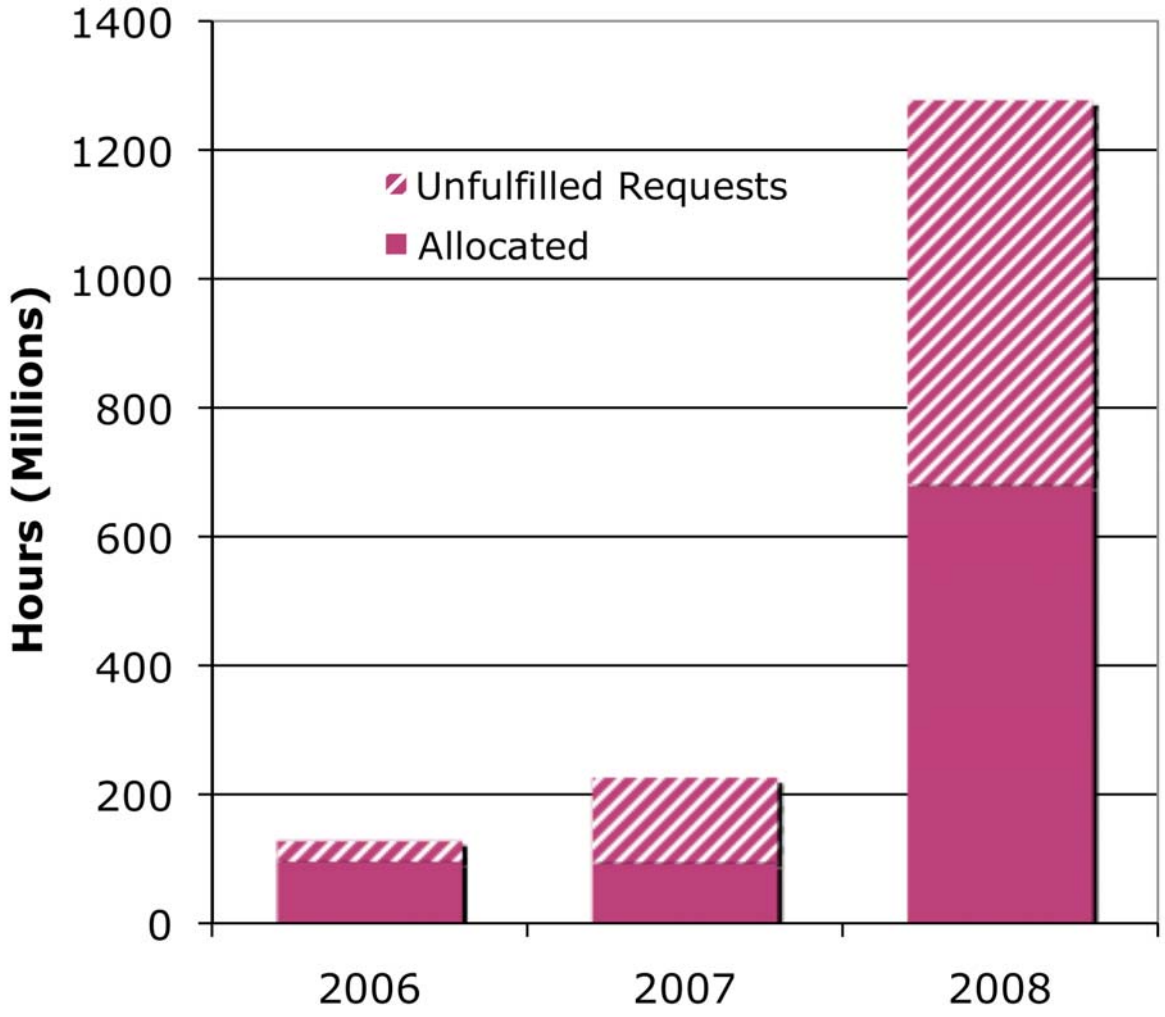


U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

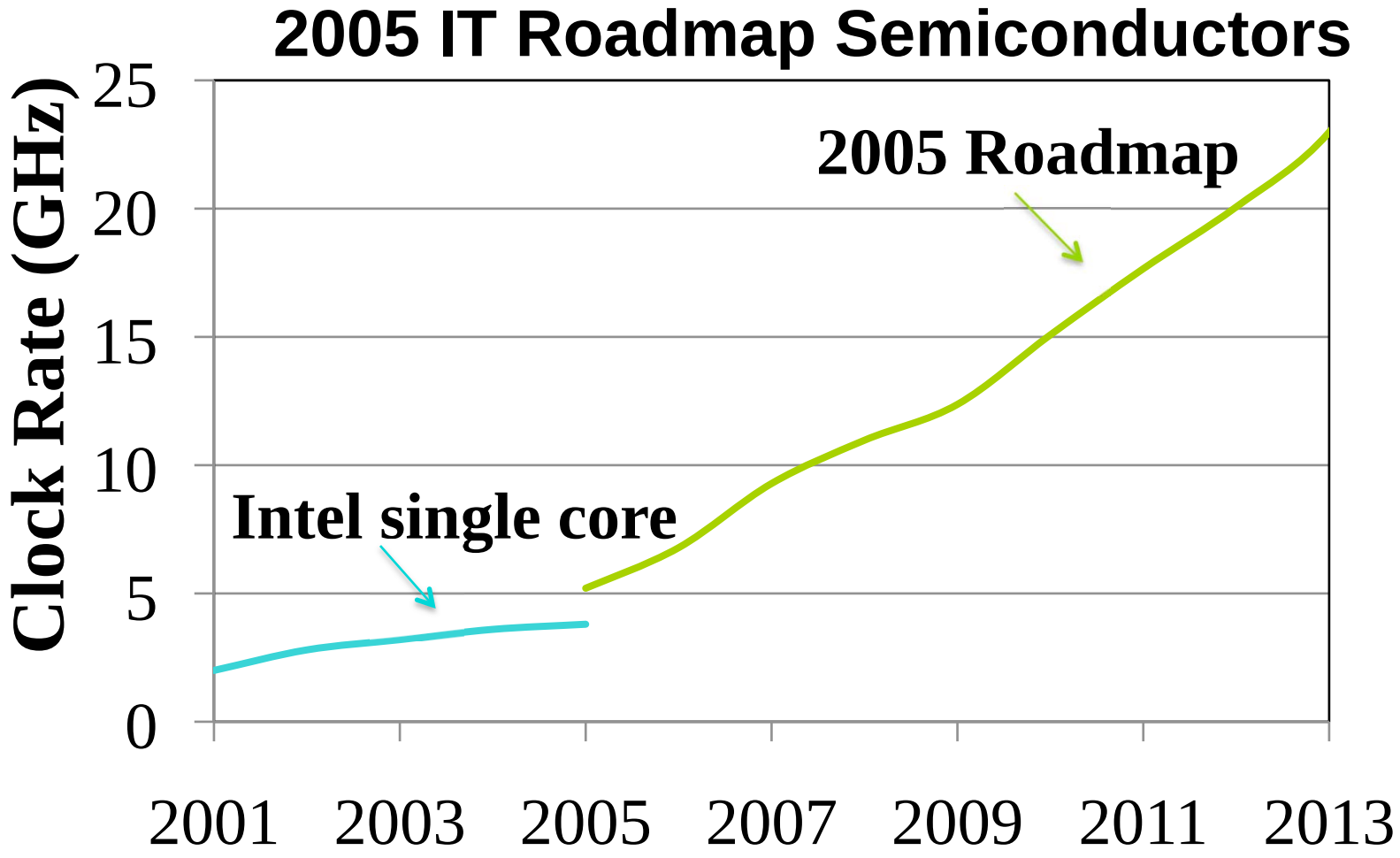
DOE Demand for Computing is Growing

Compute Hours Requested vs Allocated

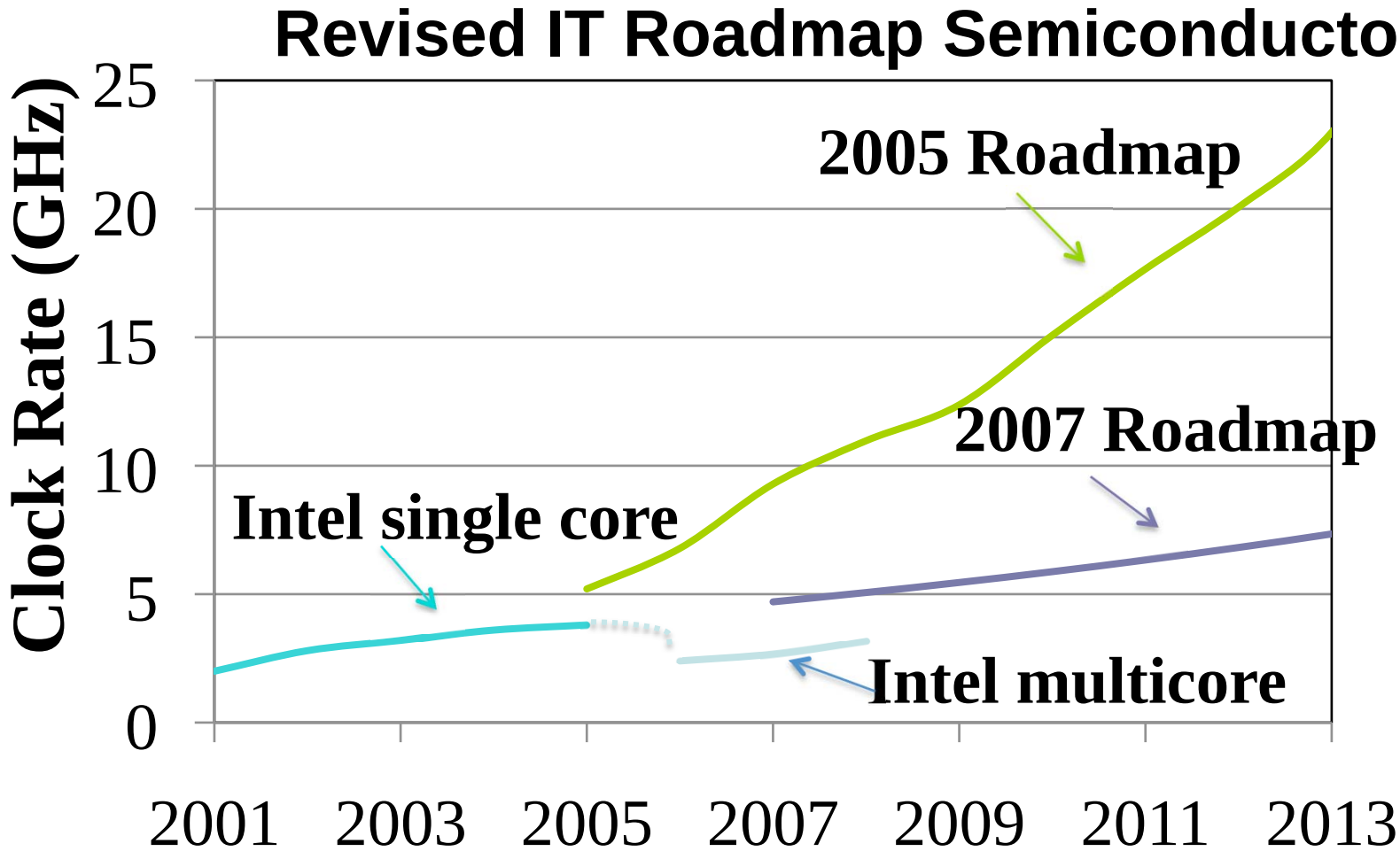


- *Each year DOE users requests 2x more hours than allocated*
- *This 2x is artificially constrained by perceived availability*
- *Unfulfilled allocation = hundreds of millions of hours in 2008*
- *When allocation limits are removed, scaling and science increase*

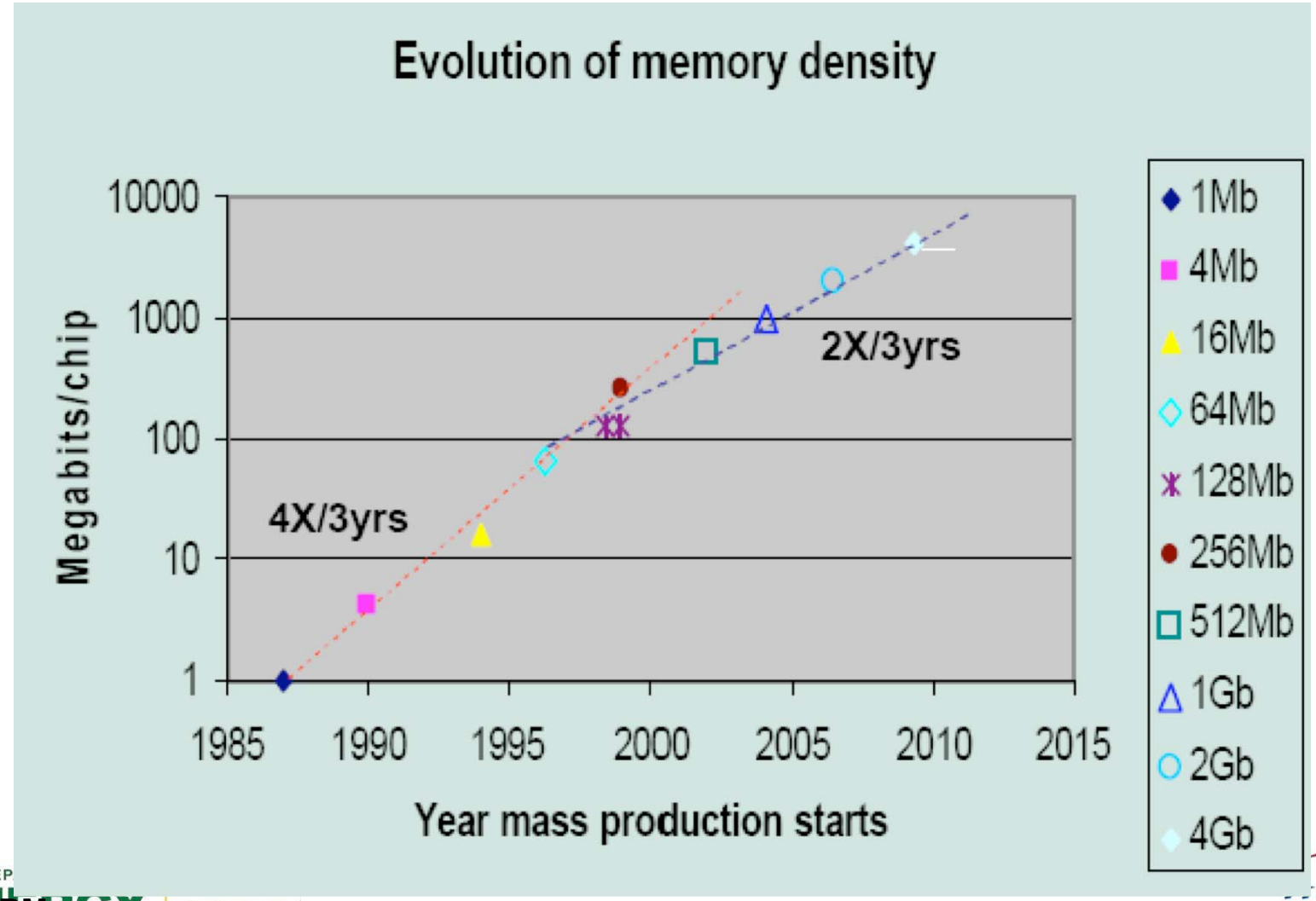
2005: Clock speed will double every 2 years



2007: Cores/chip will double every 2 years



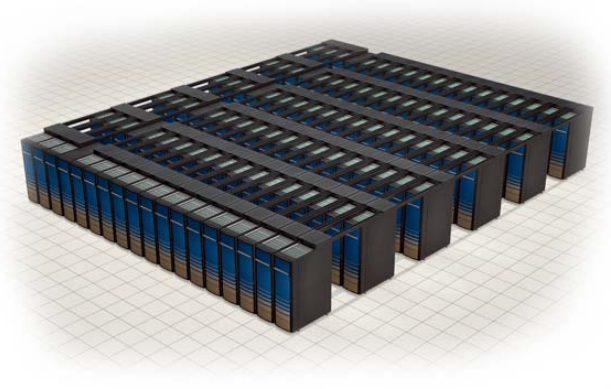
DRAM component density is only doubling every 3 years

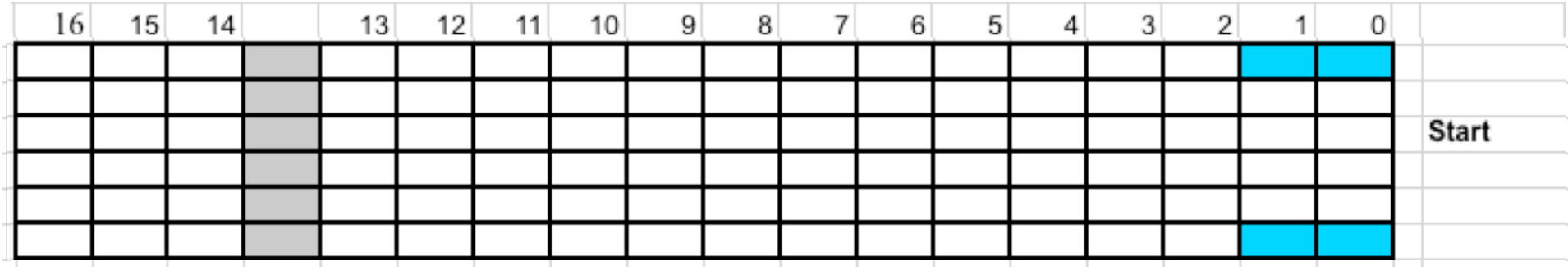


Source: IBM

New Moore's Law In Situ: Quad Core Upgrade Phases

Legend
Production DC Non-Upgraded
Production SIO DC
Empty Column
Test DC Non-Upgraded
Test SIO Module Installed QC
Test QC Upgraded
Production QC Upgraded
Production SIO Module Installed QC

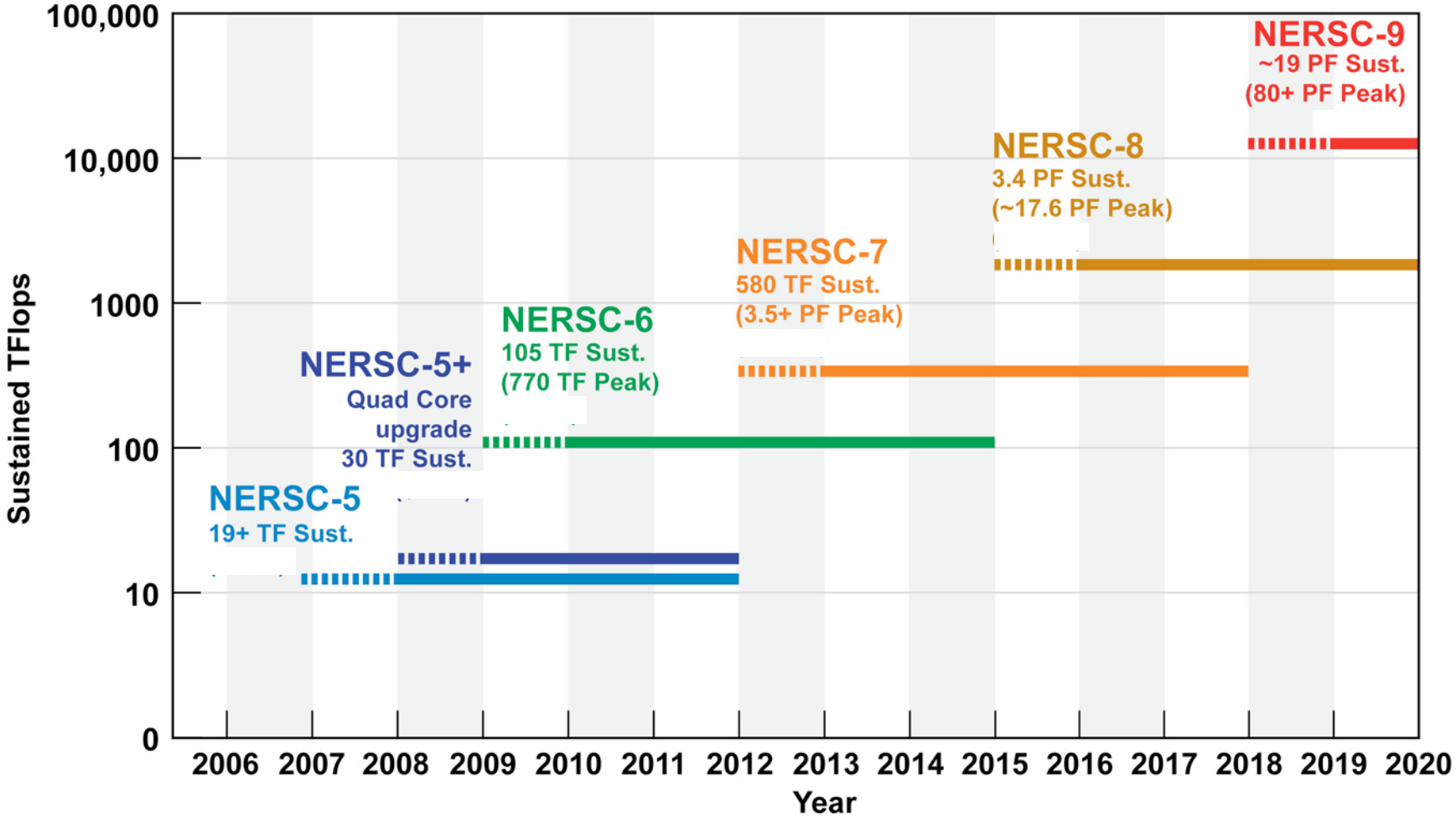




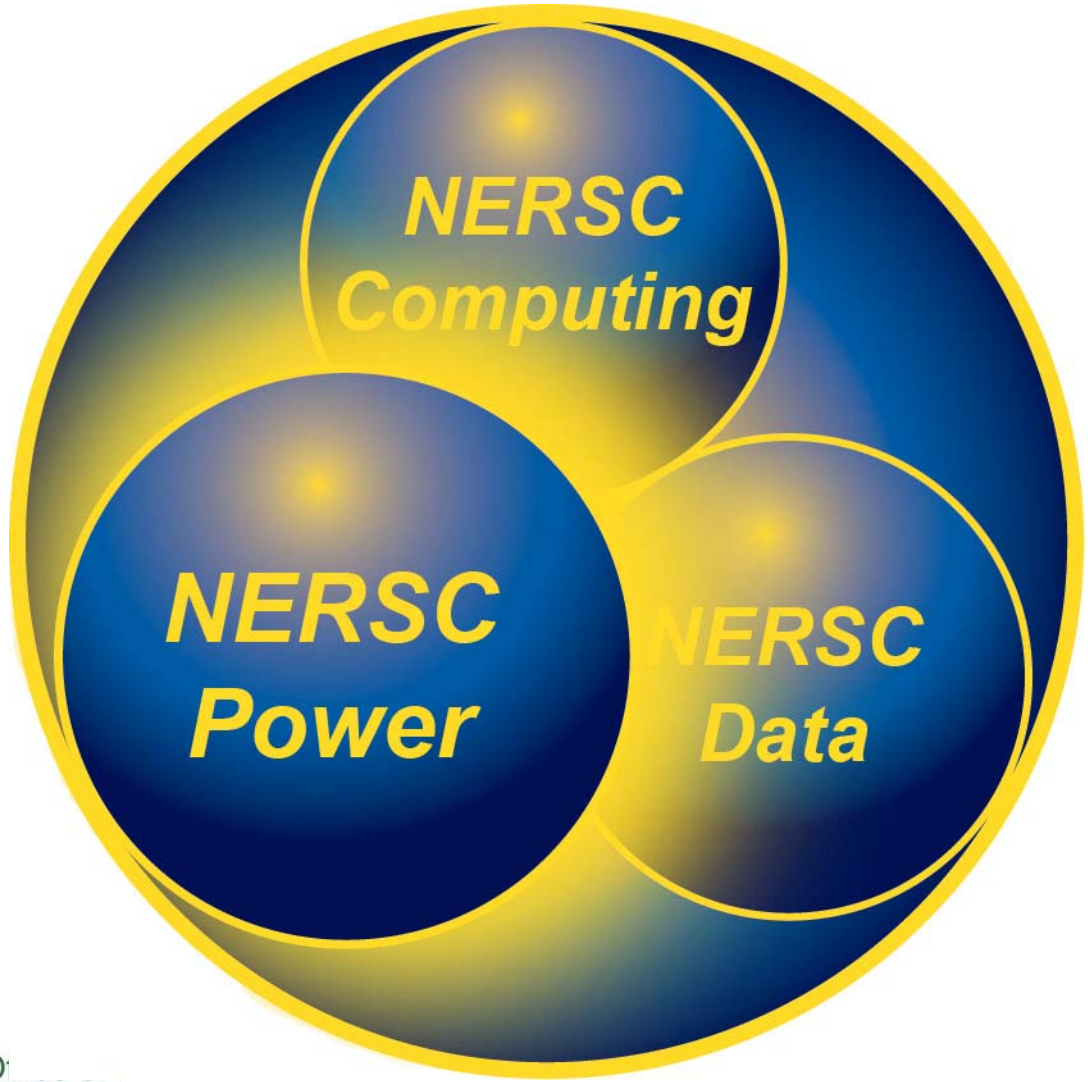
- **19,320 cores in production**

Plan developed by Bill Kramer and Nick Cardo in collaboration with Cray (Dan Unger and others).

NERSC Approximate Computational System Profile



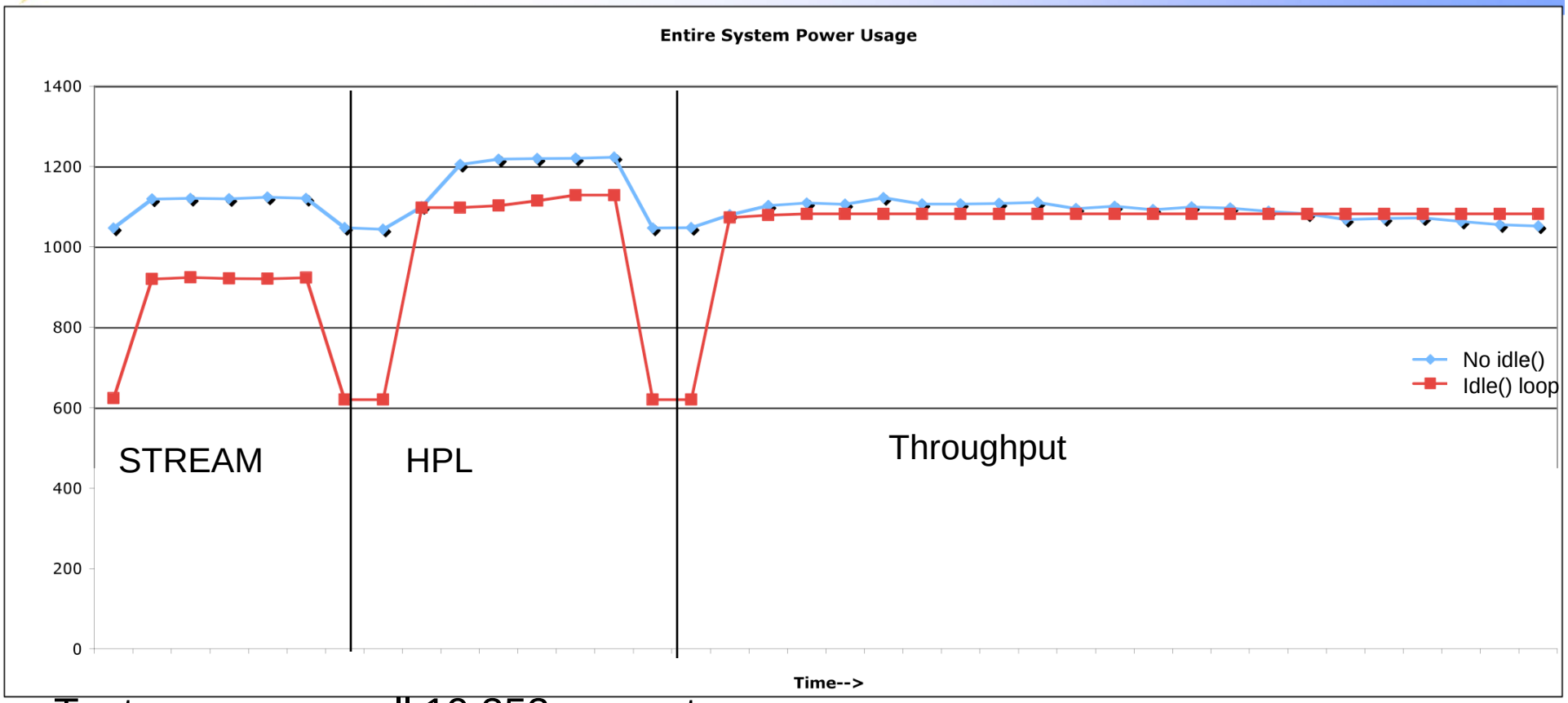
NERSC Power Efficiency



Understanding Power Consumption in HPC Systems

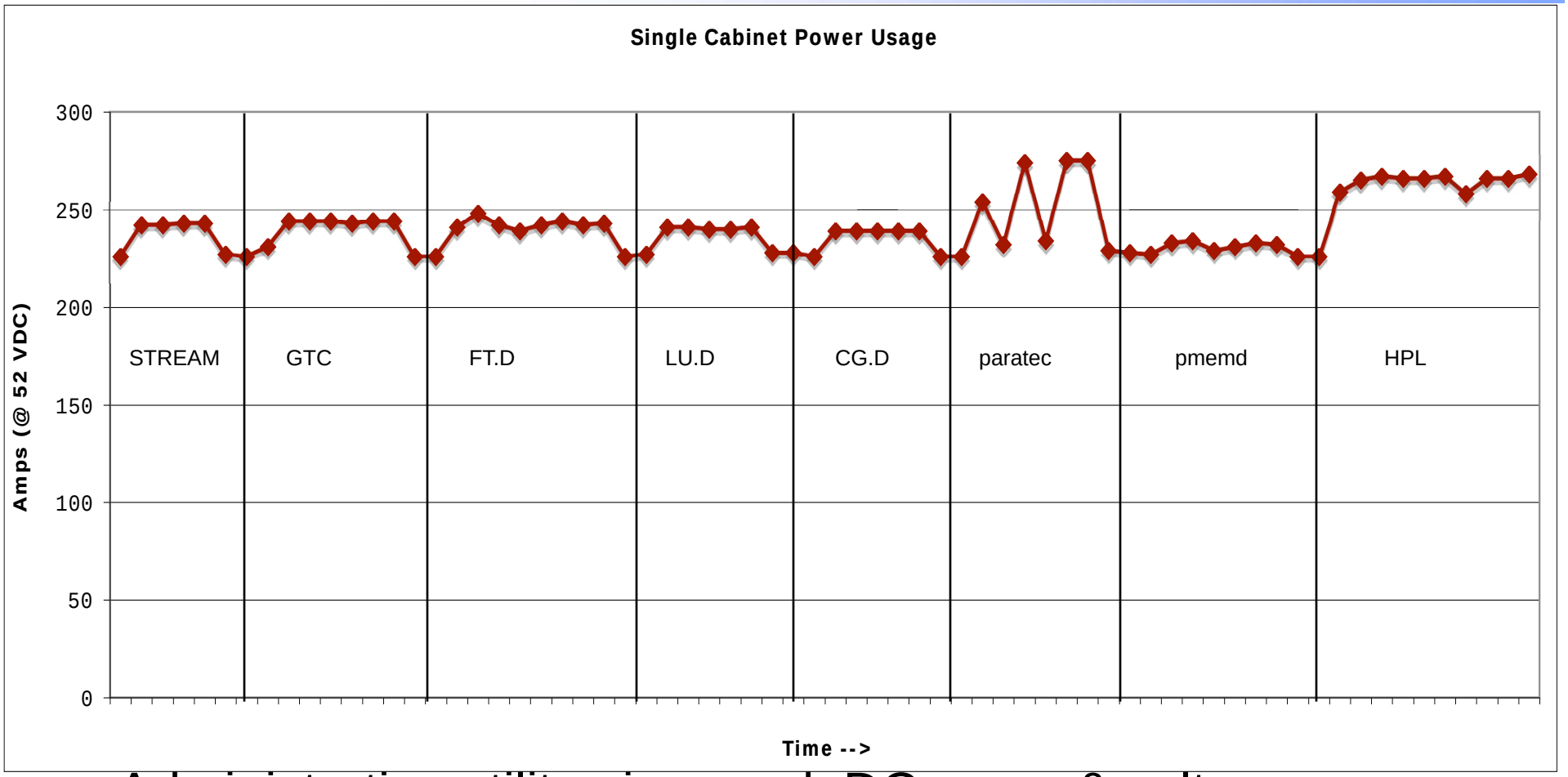
- Until about 2 - 3 years ago there has been a lack of interest in power issues in HPC
- Power is *the* barrier to reaching Exascale: projected between 20 and 200 MW
- Lack data and methodology to address power issues in computer architecture
- Project at LBNL (NERSC and CRD) to development measurement standards and better quantitative understanding

Full System Test



- Tests run across all 19,353 compute cores
- Throughput: NERSC “realistic” workload composed of full applications
- idle() loop allows powersave on unused processors; (generally more efficient)

Single Rack Tests



- Administrative utility gives rack DC amps & voltage
- HPL & Paratec are highest power usage

Power Conclusions

- **Power utilization under an HPL/Linpack load is a good estimator for power usage under mixed workloads for single nodes, cabinets / clusters, and large scale systems**
  - **Idle power is not**
  - **Nameplate and CPU power are not**
- **LINPACK running on one node or rack consumes approximately same power as the node would consume if it were part of full-sys parallel LINPACK job**
- **We can estimate overall power usage using a subset of the entire HPC system and extrapolating to total number of nodes using a variety of power measurement techniques**
  - **And the estimates mostly agree with one-another!**

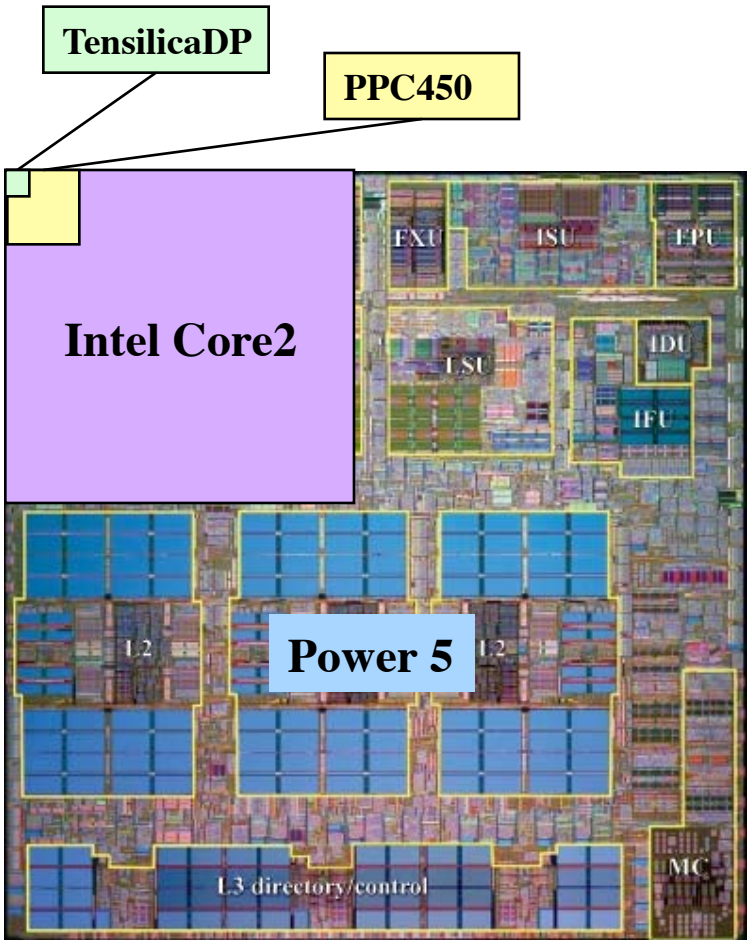
Parallelism is "Green"

"Years of research in low-power embedded computing have shown only one design technique to reduce power: reduce waste."

—Mark Horowitz, Stanford University & Rambus Inc.

- Highly concurrent systems are more power efficient
  - Dynamic power is proportional to V^2fC
  - Increasing frequency (f) also increases supply voltage (V) → more than linear effect of clock speed scaling
  - Increasing cores increases capacitance (C) but has only linearly
- High performance serial processors waste power
  - Speculation, dynamic dependence checking, etc. burn power
  - Implicit parallelism discovery
- Challenge: *Can you double the concurrency in your algorithms and software every 2 years?*

Design for Power: More Concurrency

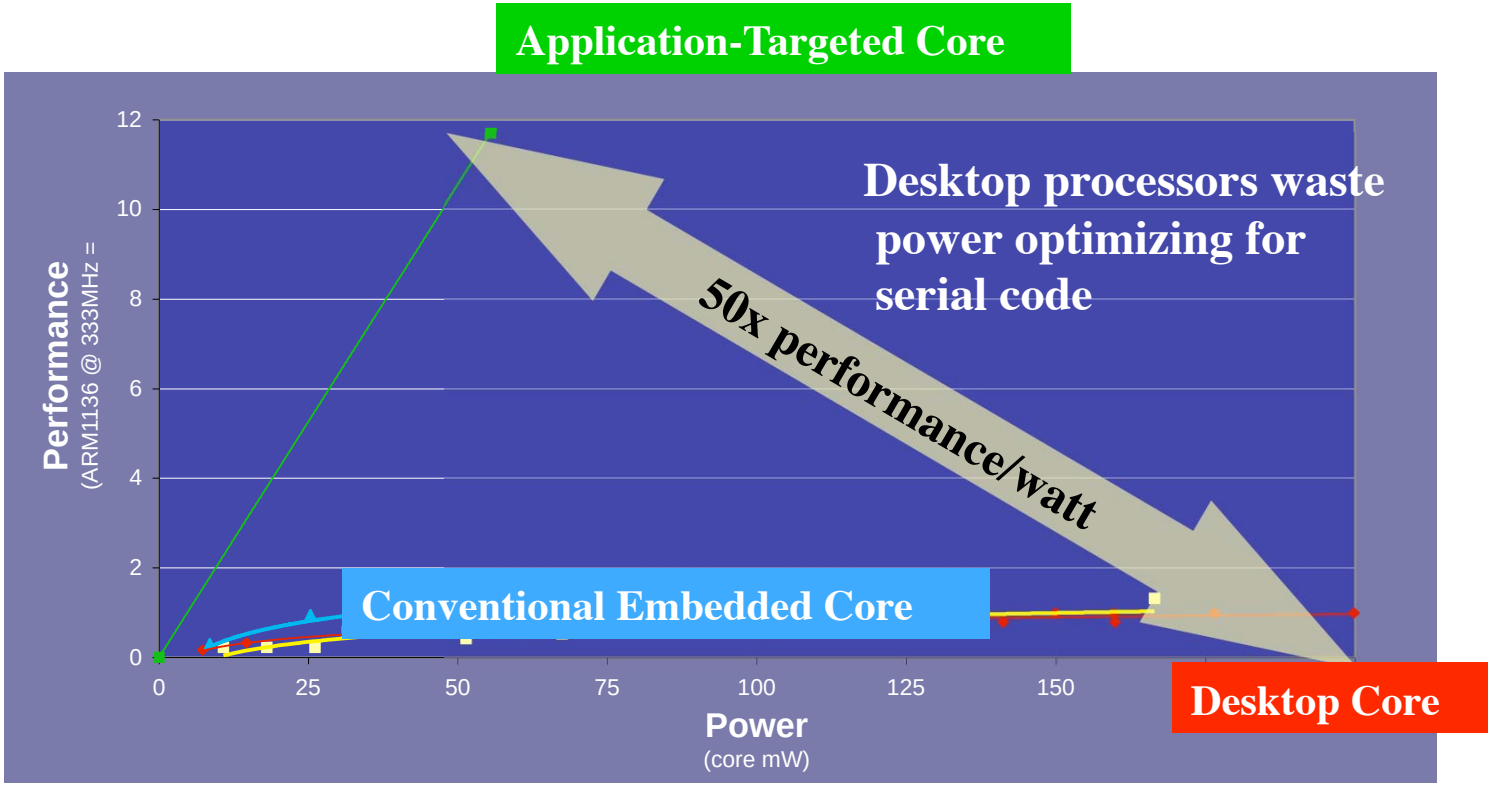


- Power5 (Server)
  - 389 mm^2
  - 120 W @ 1900 MHz
- Intel Core2 sc (Laptop)
  - 130 mm^2
  - 15 W @ 1000 MHz
- PowerPC450 (BlueGene/P)
  - 8 mm^2
  - 3 W @ 850 MHz
- Tensilica DP (cell phones)
  - 0.8 mm^2
  - 0.09 W @ 650 MHz

Each core operates at 1/3 to 1/10th efficiency of largest chip, but you can pack 100x more cores onto a chip and consume 1/20 the power!

Specialization Saves Power

Graph courtesy of Chris Rowen, Tensilica Inc.



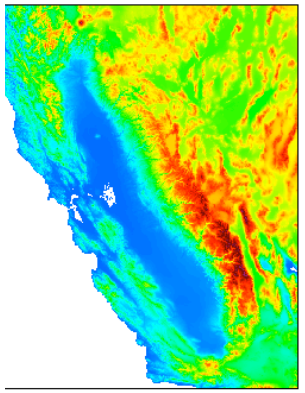
Performance on EEMBC benchmarks aggregate for Consumer, Telecom, Office, Network, based on ARM1136J-S (Freescale i.MX31), ARM1026EJ-S, Tensilica Diamond 570T, T1050 and T1030, MIPS 20K, NECVR5000). MIPS M4K, MIPS 4Ke, MIPS 4Ks, MIPS 24K, ARM 968E-S, ARM 966E-S, ARM926EJ-S, ARM7TDMI-S scaled by ratio of Dhrystone MIPS within architecture family. All power figures from vendor websites, 2/23/2006.

1km-Scale Global Climate Model Requirements

1km-Scale required to resolve clouds

- Simulate climate 1000x faster than real time
- 10 Petaflops sustained per simulation (~200 Pflops peak)
- 10-100 simulations (~20 Exaflops peak)
- DOE E3SGS report suggests exaflop requires 180MW

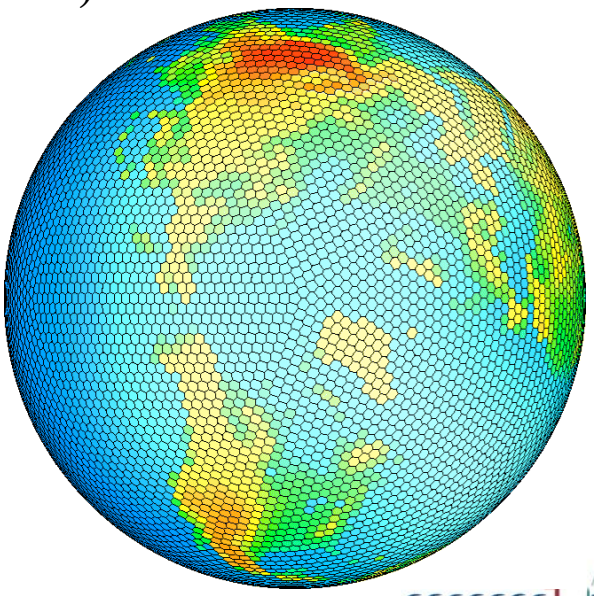
200km (now)



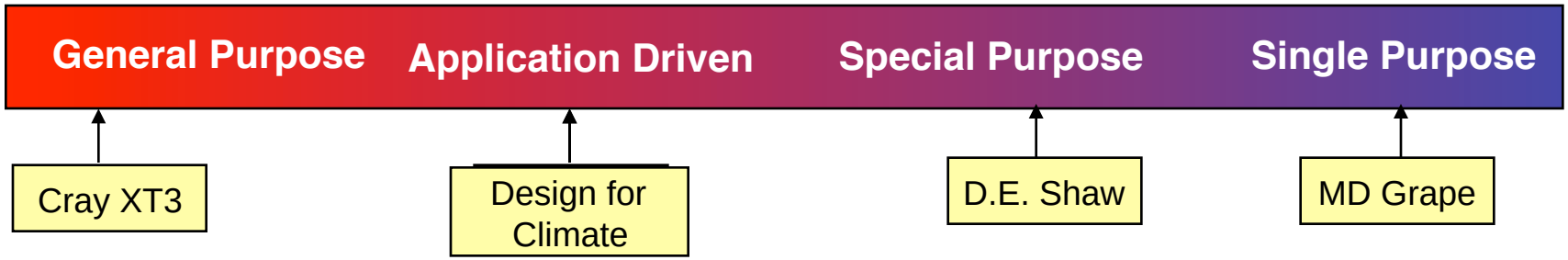
1km

Computational Requirements:

- Advanced dynamics algorithms: icosahedral, cubed sphere, reduced mesh, etc.
- ~20 billion cells → 100 Terabytes of Memory
- Decomposed into ~20 million total subdomains → massive parallelism

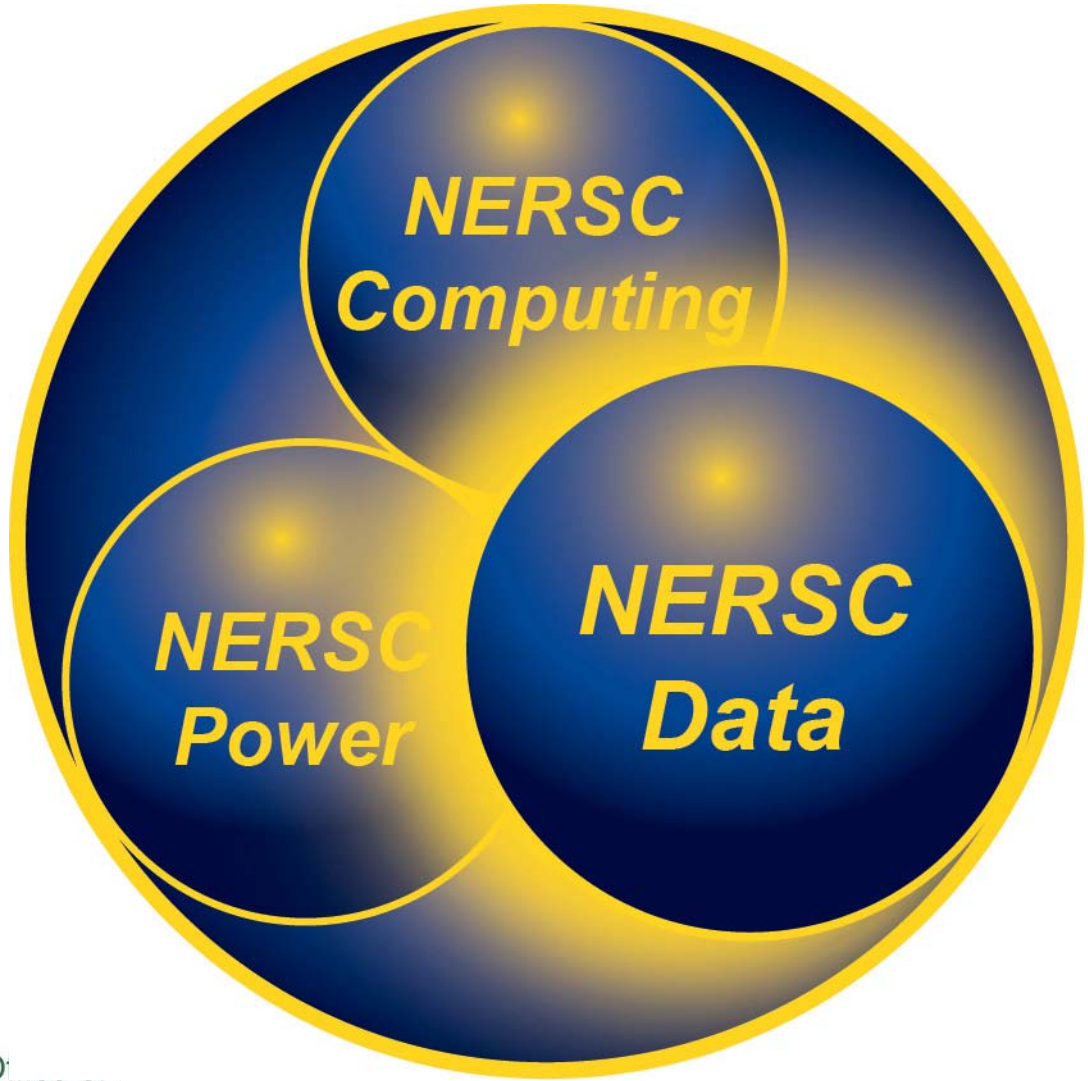


Strawman 1km Climate Computer
(Shalf, Oliker, Wehner)

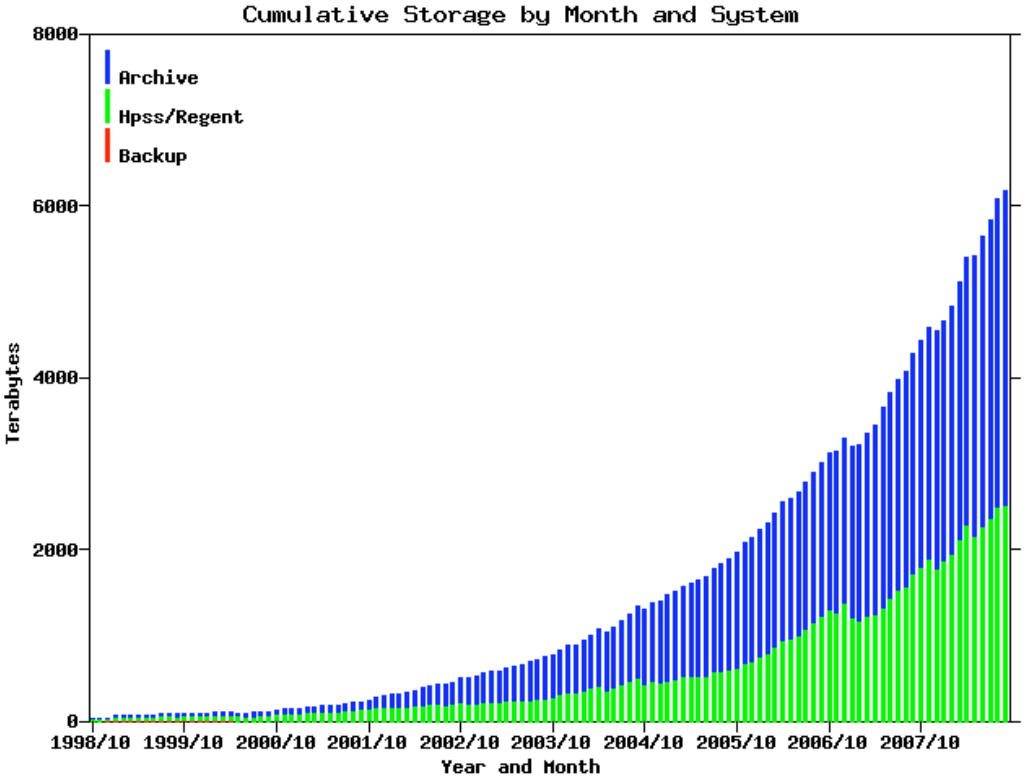


- Computation
  - .015ºX.02ºX100L (note 4X more vertical levels than CCSM3)
- Hardware:
  - ~10 Petaflops sustained (300 Pflops peak?); ~100 Terabytes total memory
  - ~20 million processors using true commodity (embedded cores)
- Massively parallel algorithms with autotuning
  - E.g., scalable data structure, e.g., Icosahedral with 2D partitioning
  - ~20,000 nearest neighbor communication pairs per subdomain per simulated hour of ~10KB each
- Upside result:
  - 1K scale model running in O(5 years)! 10-100x less energy.
- Worse case:
  - Better understand how to build systems & algorithms for climate

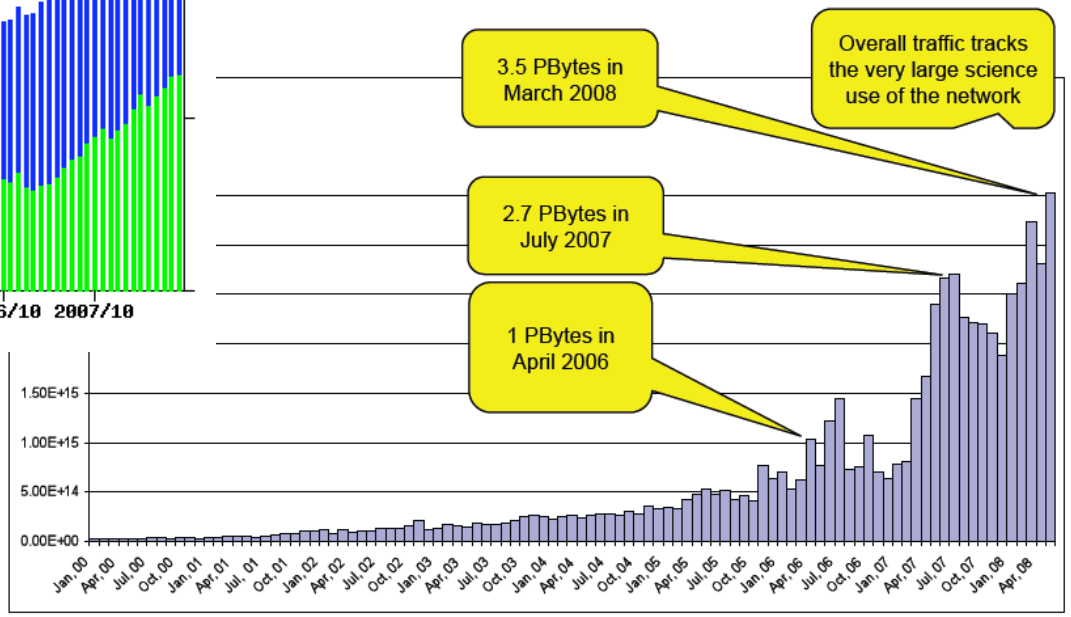
NERSC Data



Data is Increasing Faster than Moore's Law



NERSC Archive increasing at 70% GAGR

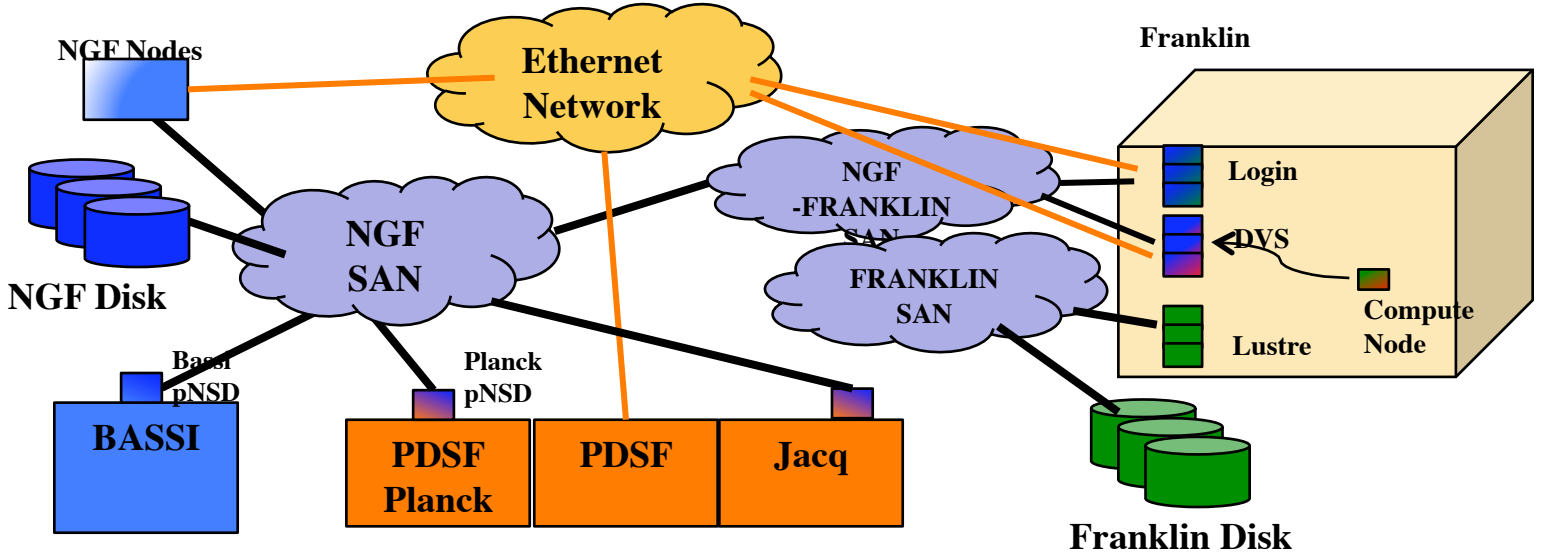


ESnet traffic historically increasing at 80% CAGR

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

NERSC's Global File System (NGF)
The First of Its Kind



- A facility-wide file system
  - Scientists more productive; efficient use of unique computational resources
  - Integration with archival storage (more desired) and grid desired
- High performance
  - Scales with clients and storage devices
  - Absolute performance close to that of local parallel file systems

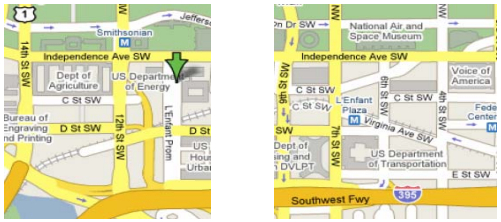
NERSC Data Elements: Tools

- New Paradigm for Analytics: “Google for Science”
- MapReduce: A simple programming model that applies to many large-scale analytics problems
- Hide messy details in MapReduce runtime library:
  - Parallelization, load balancing, machine failures, …
- Steps in MapReduce:
  - Read a lot of data
  - Map: extract interesting items
  - Shuffle and Sort
  - Reduce: aggregate, transform,…
  - Write the results
- Used at Google for >10K applications
  - Grep, clustering, machine learning, …

List of roads, intersections,…

Find those in given lat/long range

Render map tiles



U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Multicore Technology Summary

- Multicore sustains Moore's Law growth
  - Memory wall issues continue to rise
  - Data storage needs will continue to rise
- Multicore helps power issues
  - On-chip power density; total system power
- Architectural chaos:
  - What is a "core"?
    - 1 thread per core, many threads per core (Niagra, XMT), many "cores" per thread (vectors, SIMD, …)
- Software challenges are key

Strategies for Multicore (and Manycore) in Exascale

- There are multiple approaches we could take in the HPC community
- They have different cost to us:
  - Software infrastructure investment
  - Application software investment
- And different risks of working
  - At all
  - Or at the performance level we demand

1) MPI Everywhere

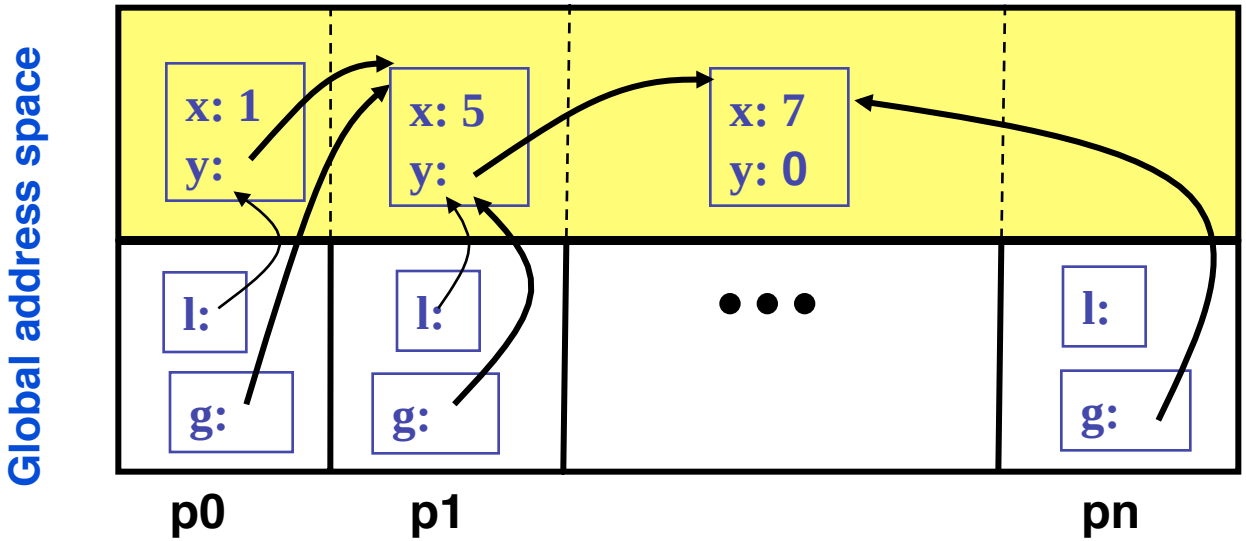
- We can run 1 MPI process per core
  - This works now (for CMPs) and will work for a while
- How long will it continue working?
  - 4 - 8 cores? Probably. 128 - 1024 cores? Probably not.
  - Depends on performance expectations -- more on this later
- What is the problem?
  - Latency: some copying required by semantics
  - Memory utilization: partitioning data for separate address space requires some replication
    - How big is your per core subgrid? At 10x10x10, over 1/2 of the points are surface points, probably replicated
  - Memory bandwidth: extra state means extra bandwidth
  - Weak scaling will not save us -- not enough memory per core
  - Heterogeneity: MPI per CUDA thread-block?
- Advantage: no new apps work; modest infrastructure work (multicore-optimized MPI)

2) Mixed MPI and OpenMP

- This is the obvious next step
- Problems
  - Will OpenMP performance scale with the number of cores / chip?
  - OpenMP does not support locality optimizations
  - More investment in infrastructure than MPI, but can leverage existing technology
  - Heterogeneity support unclear
  - Do people want two programming models?
- Advantages
  - Incremental work for applications
- Variation: await a silver bullet from industry
  - Will this be at all helpful in scientific applications?
  - Do they know enough about parallelism/algorithms

3) PGAS Languages

- *Global address space:* **thread may directly read/write remote data**
- *Partitioned:* **data is local or global: critical for scaling**
  - **Maps directly to shared memory hardware**
  - **Maps to one-sided communication on distributed memory hardware**
  - **One programming model for inter and intra node parallelism!**



- *UPC, CAF, Titanium: Static parallelism (1 thread per proc)*
  - **Does not virtualize processors; main difference from HPCS languages which have many/dynamic threads**

Sharing and Communication Models: PGAS vs. Threads

- "Shared memory" OpenMP, Threads,…
  - No control over locality

  ⇒Caching (automatic management of memory hierarchy) is critical

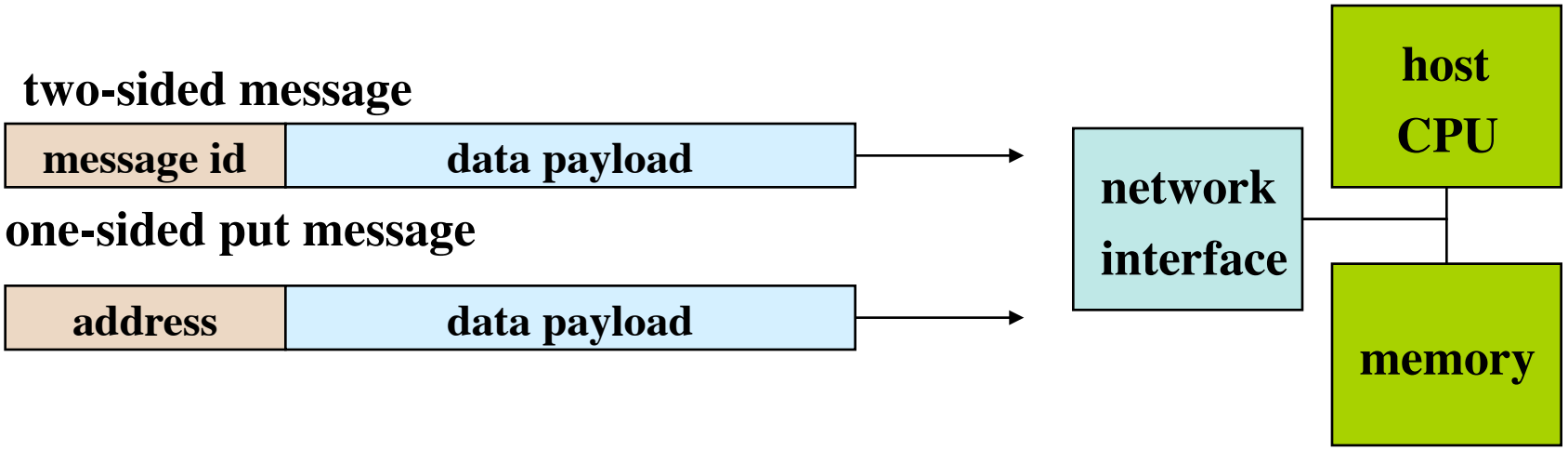
  ⇒Cache coherent needed (hw or sw)
- PGAS / One-sided Communication
  - Control over locality, explicit movement

  ⇒Caching is not required; programmer makes local copies and manages their consistency

  ⇒Need to read/write without bothering remote application (progress thread, DMA)

  ⇒No cache coherent needed, except between the network interface and procs in a node

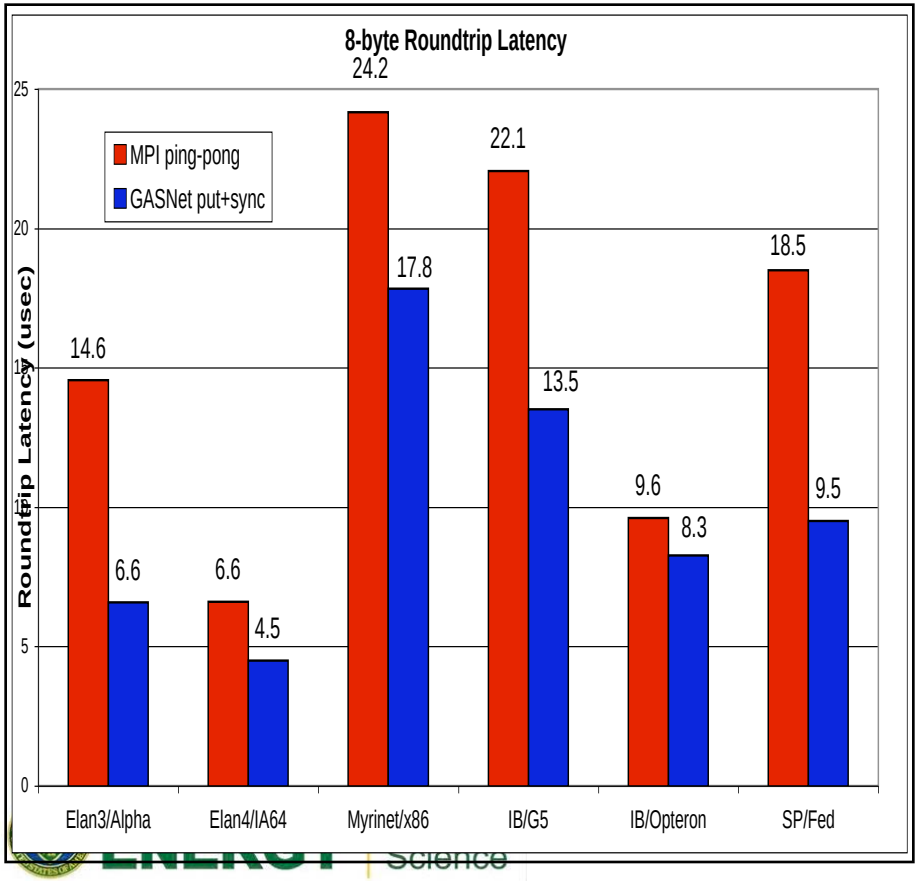
Sharing and Communication Models: PGAS vs. MPI

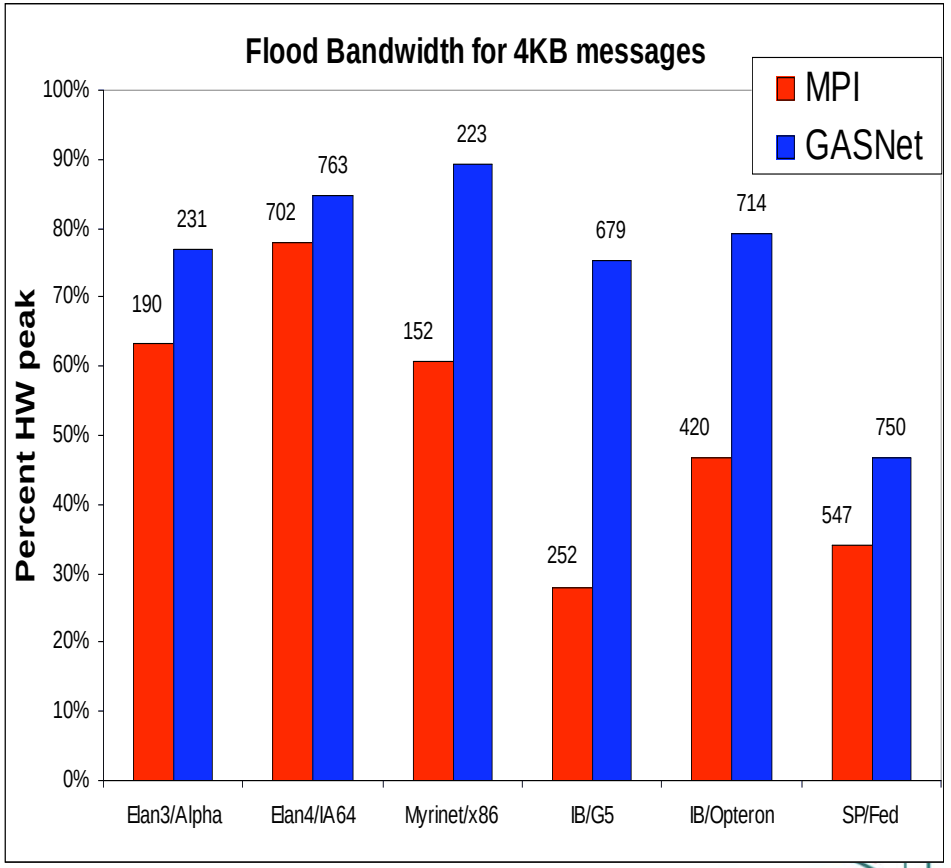


- A one-sided put/get message can be handled directly by a network interface with RDMA support
  - Avoid interrupting the CPU or storing data from CPU (preposts)
- A two-sided messages needs to be matched with a receive to identify memory address to put data
  - Offloaded to Network Interface in networks like Quadrics
  - Need to download match tables to interface (from host)

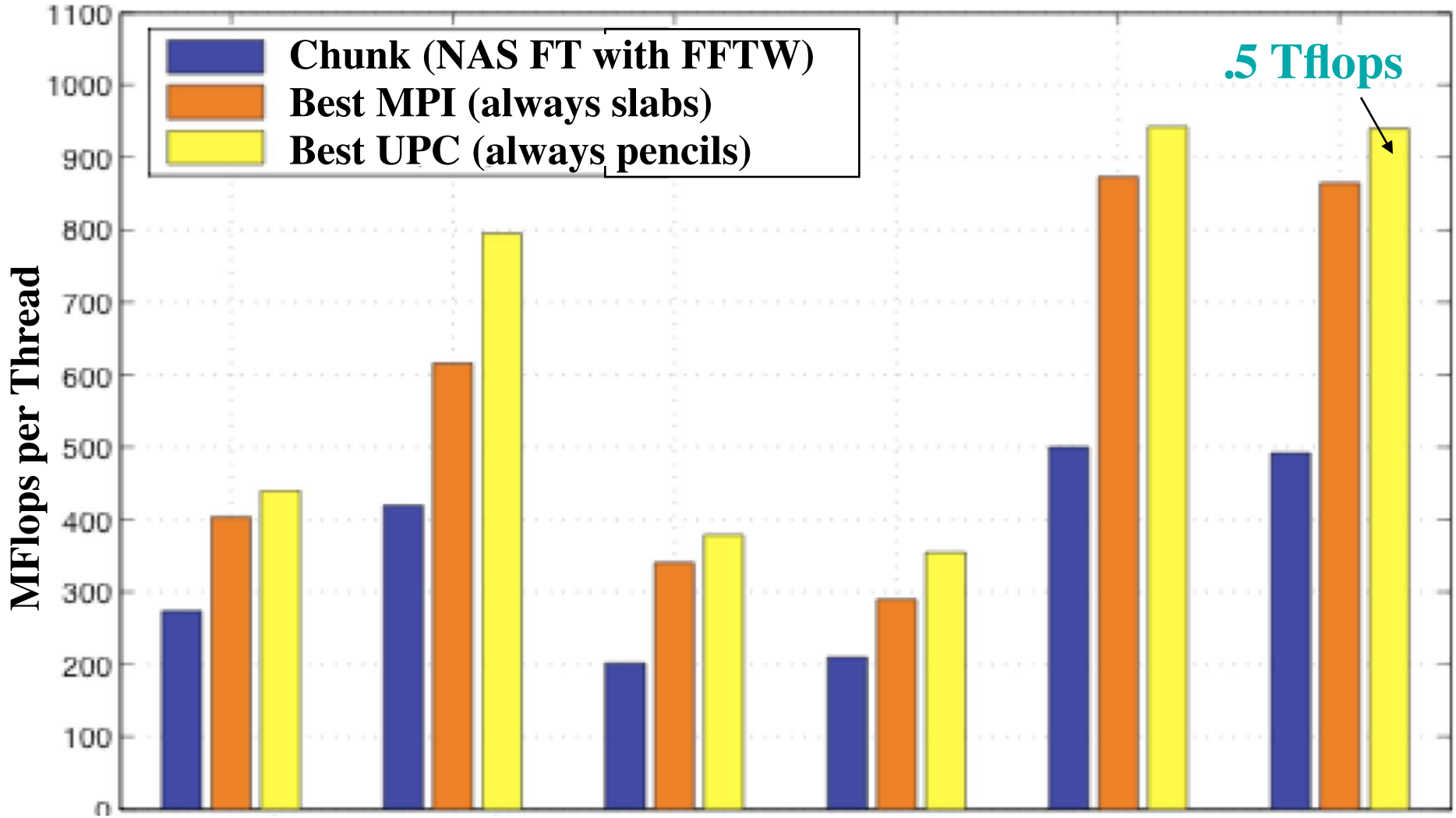
Performance Advantage of One-Sided Communication

- The put/get operations in PGAS languages (remote read/write) are one-sided (no required interaction from remote proc)
- This is faster for pure data transfers than two-sided send/receive

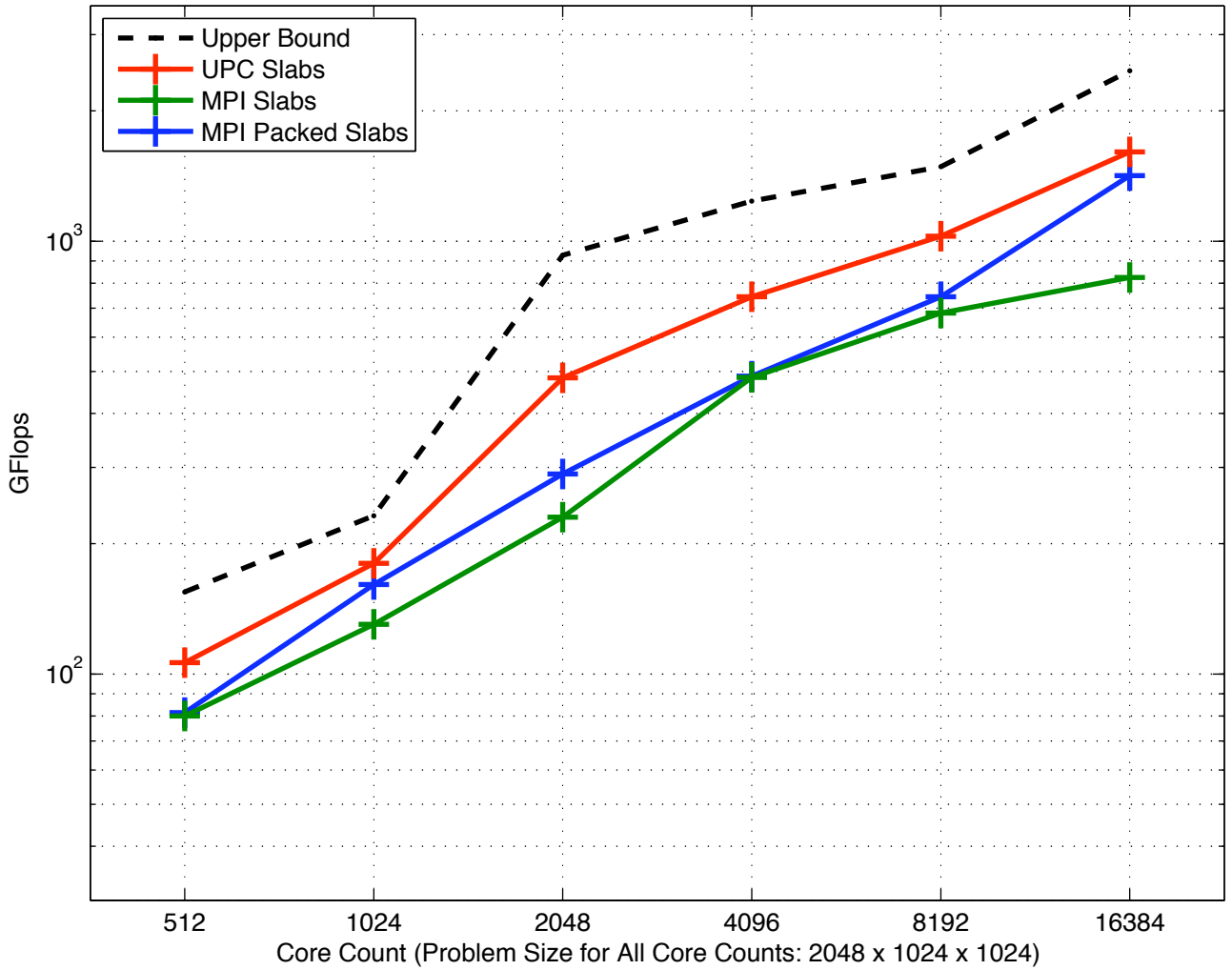




NAS FT Variants Performance Summary



3D FFT on BG/P


Upper Bound
UPC Slabs
MPI Slabs
MPI Packed Slabs
GFlops
10^3
10^2
512
1024
2048
4096
8192
16384
Core Count (Problem Size for All Core Counts: 2048 x 1024 x 1024)

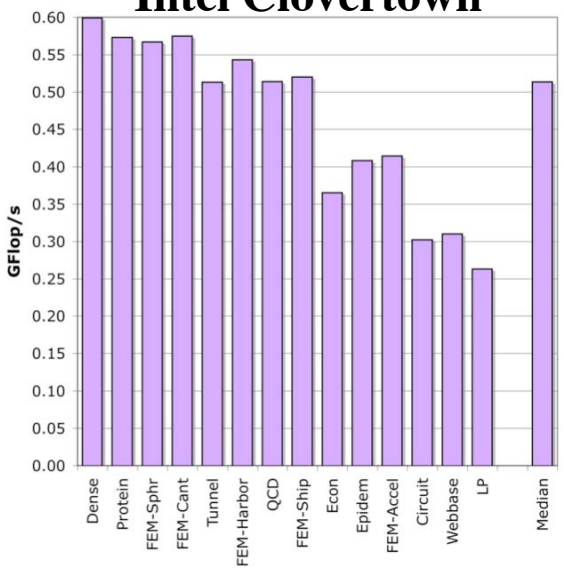
Autotuning for Multicore: Extreme Performance Programming

- Automatic performance tuning
  - Use machine time in place of human time for tuning
  - Search over possible implementations
  - Use performance models to restrict search space
- Programmers should write programs to generate code, not the code itself
- Autotuning finds a good performance solution be heuristics or exhaustive search
  - Perl script generates many versions
  - Generate SIMD-optimized kernels
  - Autotuner analyzes/runs kernels
  - Uses search and heuristics
- Can do this in libraries (Atlas, FFTW, OSKI) or compilers (ongoing research)

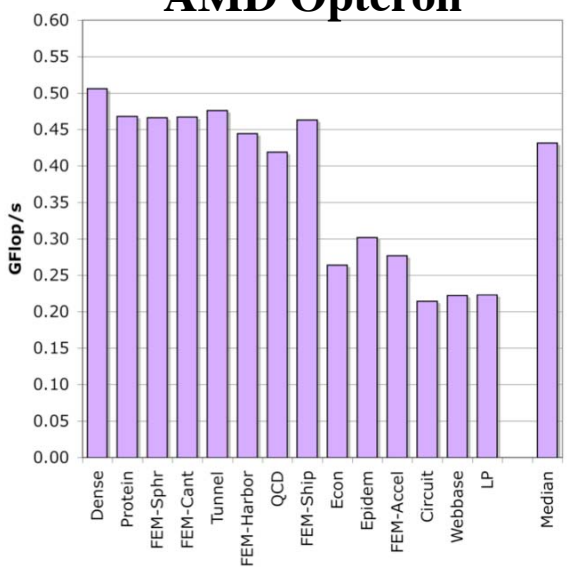
Naïve Serial Implementation

NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

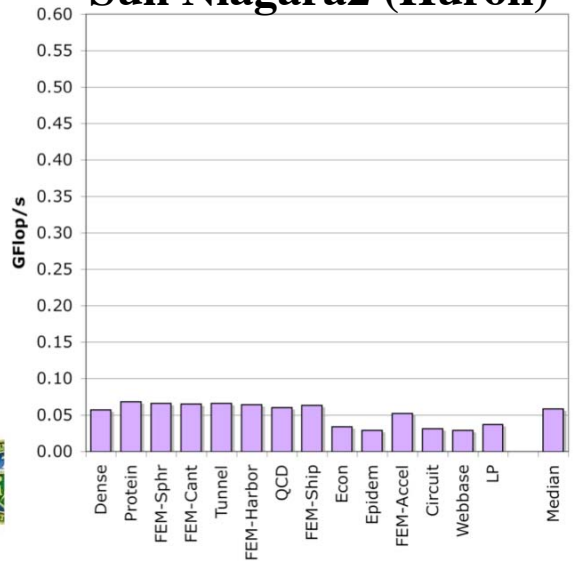
Intel Clovertown



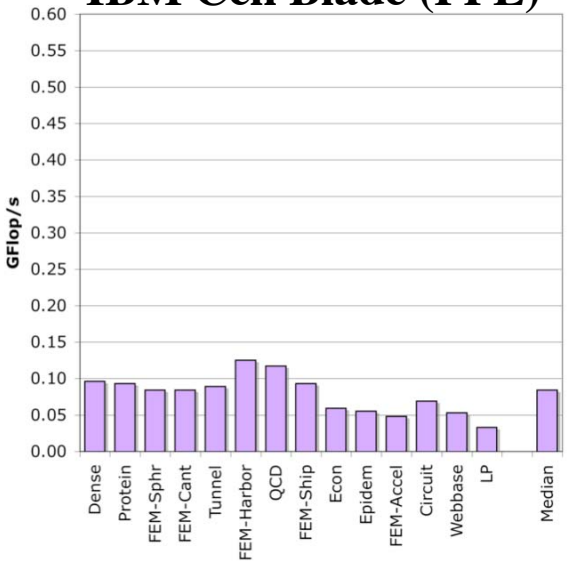
AMD Opteron



Sun Niagara2 (Huron)



IBM Cell Blade (PPE)



- Vanilla C implementation
- Matrix stored in CSR (compressed sparse row)
- Explored compiler options, but only the best is presented here
- x86 core delivers > 10x the performance of a Niagara2 thread

- Work by Sam Williams with Vuduc, Oliker, Shalf, Demmel, Yelick

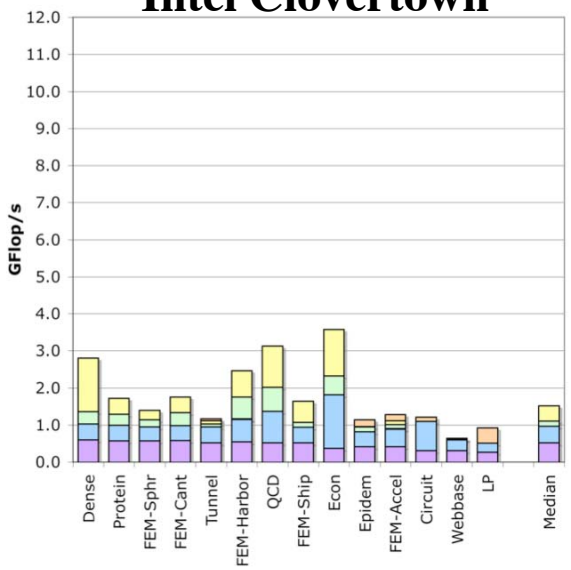
BERKELEY LAB

Autotuned Performance

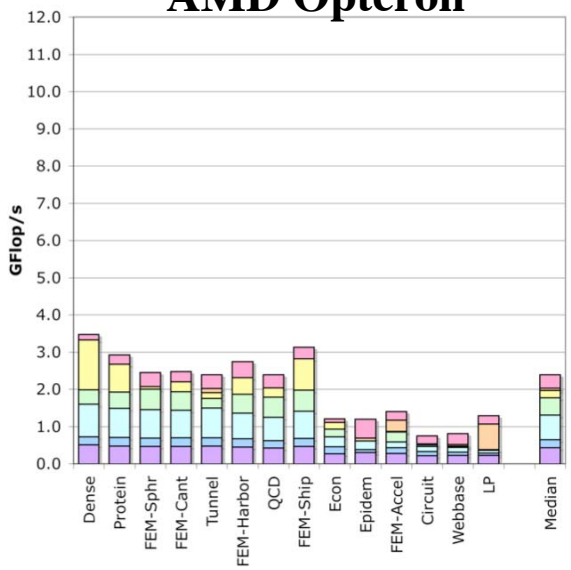
(+Cell/SPE version)

NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

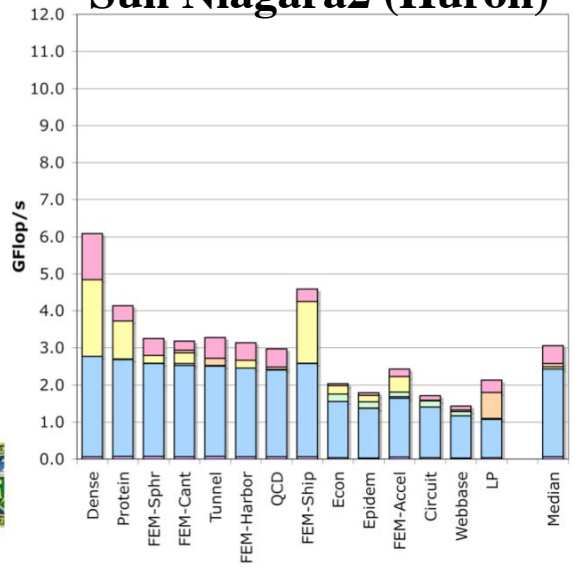
Intel Clovertown



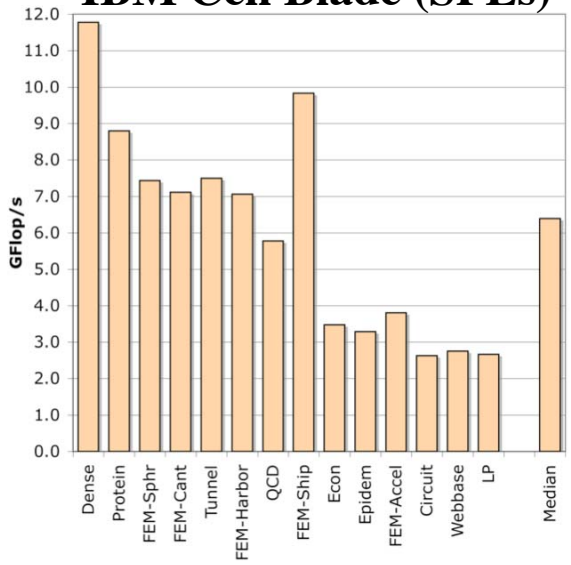
AMD Opteron



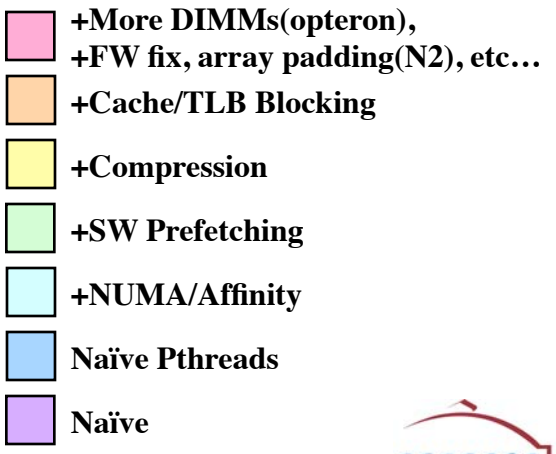
Sun Niagara2 (Huron)



IBM Cell Blade (SPEs)

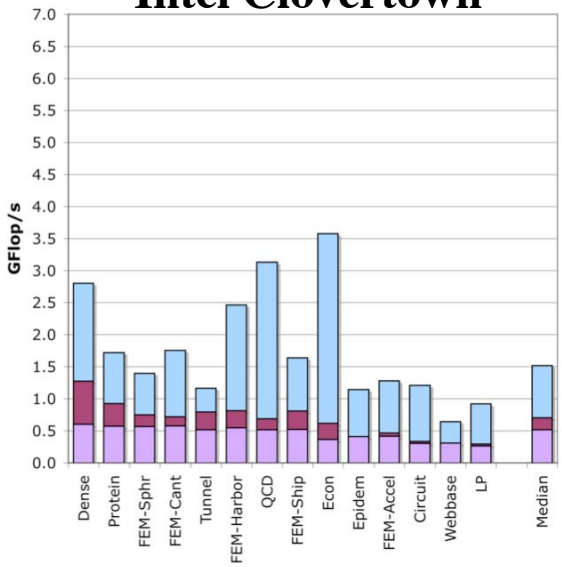


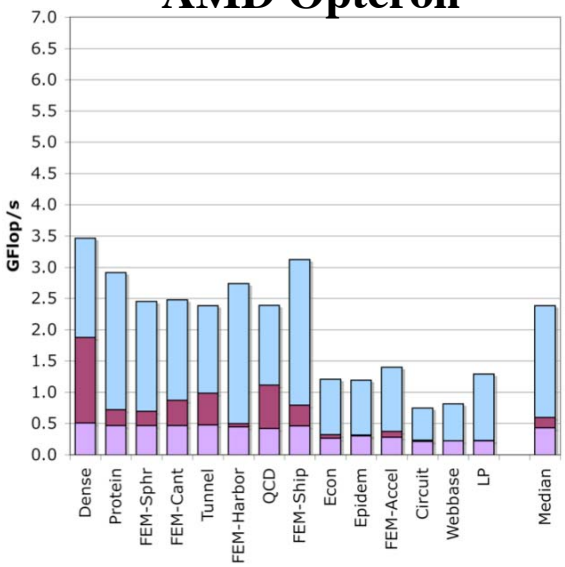
- Wrote a double precision Cell/SPE version
- DMA, local store blocked, NUMA aware, etc…
- Only 2x1 and larger BCOO
- Only the SpMV-proper routine changed
- About 12x faster (median) than using the PPEs alone.

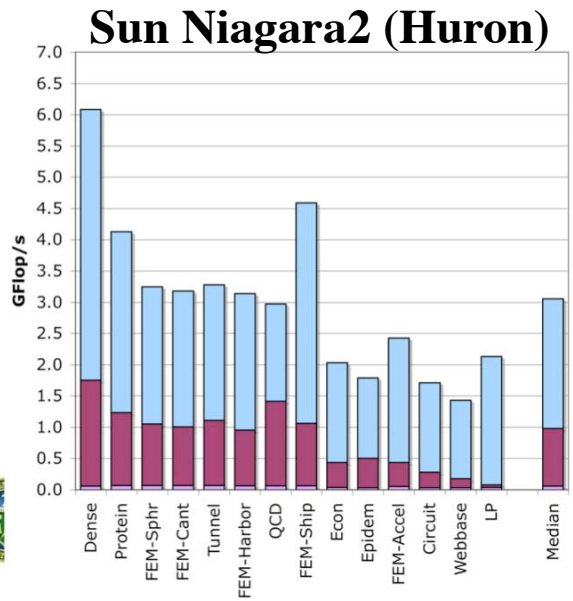


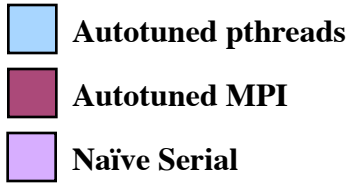
BERKELEY LAB

MPI vs. Threads









- On x86 machines, autotuned shared memory MPICH implementation rarely scales beyond 2 threads
- Still debugging MPI issues on Niagara2, but so far, it rarely scales beyond 8 threads.

Lessons Learned

- Given that almost all future scaling will be increasing cores (within or between chips), parallel programs must be more efficient than ever
- PGAS languages offer a potential solution for both
  - One-sided communication is faster than 2-sided
  - FFT example shows application level benefit
  - Allow sharing of data structures for poor memory scaling
  - Allow locality control for multisocket and multinode systems
- Autotuning promising for specific optimizations
  - Kernels within a single multicore (multisocket) node
  - Parallel libraries like collectives

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Questions?

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB