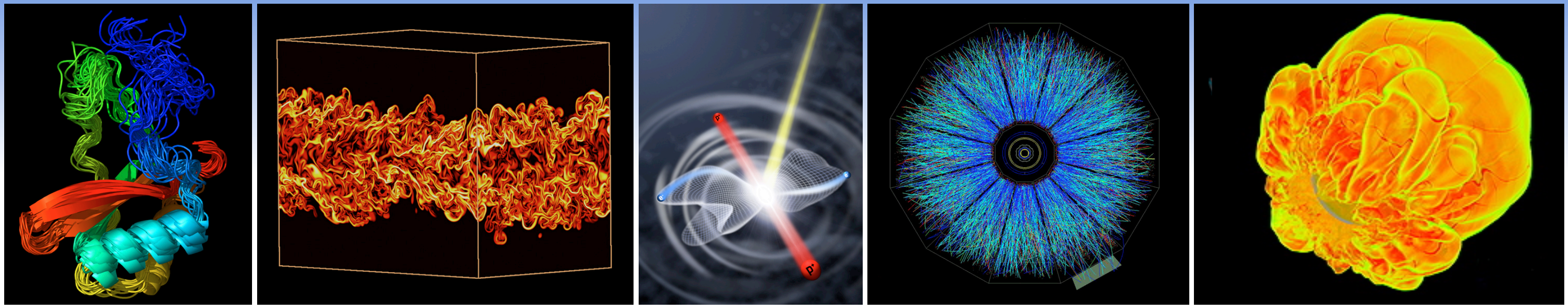


Future Scientific Computing Challenges at NERSC

Harvey Wasserman

NERSC Science Driven System Architecture Group

University of Southampton
September 30, 2008

Office of Science
U.S. DEPARTMENT OF ENERGY

NERSC

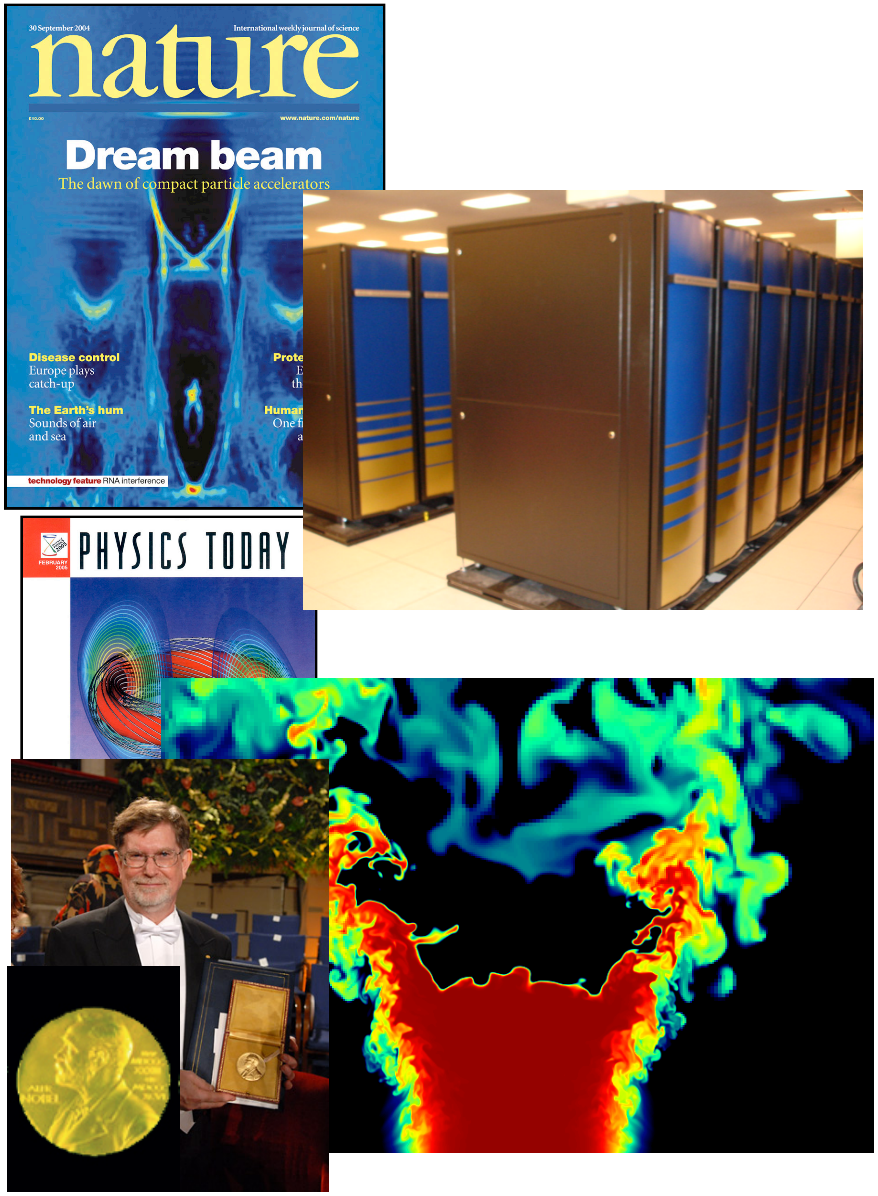
BERKELEY LAB

About Berkeley Laboratory

- Lawrence Berkeley National Laboratory
  - Located above U.C. Berkeley campus
  - DOE Office of Science (SC) program

- Research Areas (http://www.lbl.gov)
  - Nanomaterials, Particle Physics / Particle Accelerators, Astrophysics / Astronomy / Cosmology, Energy Efficiency
  - Computer Science
    - Computational Research Division (CRD), ESNet
    - National Energy Research Supercomputing Center (NERSC)

About NERSC



- Flagship user facility for all DOE Office of Science users.
- Mission: accelerate pace of scientific discovery by providing high performance computing, information, data, and communications services.
- Provide a stable production environment to deliver these services.

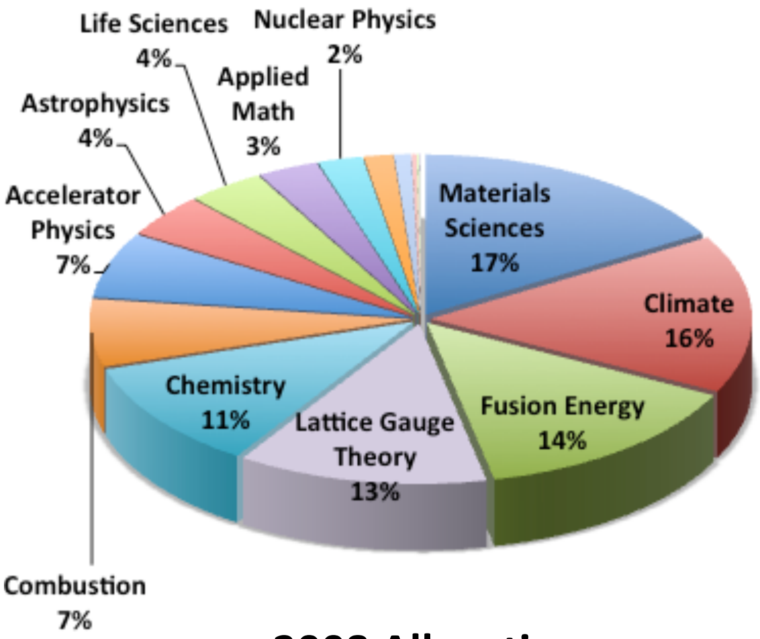
ISO-Base™ Seismic Isolation Platform

NERSC

NATIONAL ENERGY RESEARCH
SCIENTIFIC COMPUTING CENTER

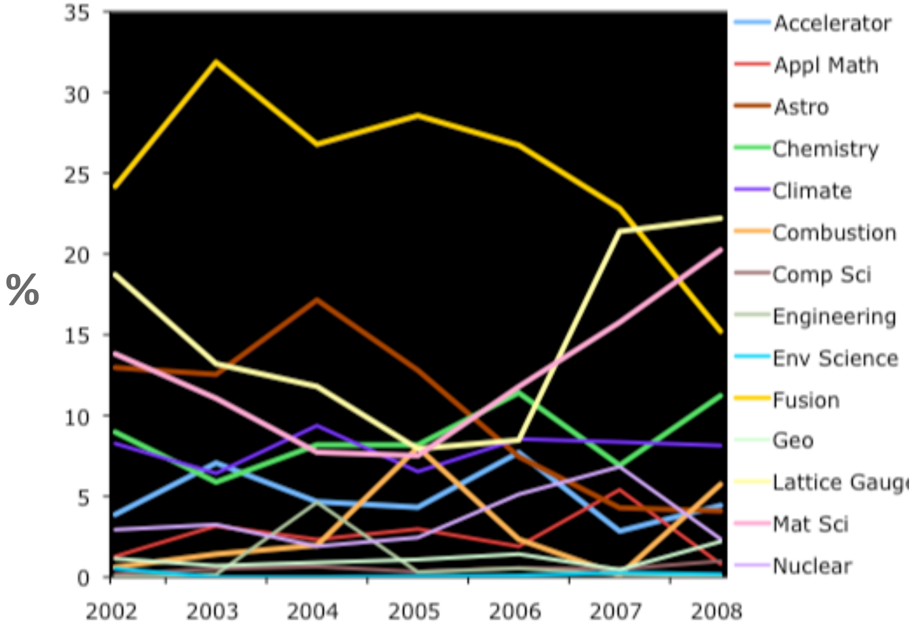
NERSC Workload

- ~3000 users, ~400 projects nationwide



2008 Allocations

Usage by Science Area as a Percent of Total Usage



Office of Science
U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

How Science Drives Architecture

Algorithm Science areas	Dense linear algebra	Sparse linear algebra	Spectral Methods (FFTs)	Particle Methods	Structured Grids	Unstructured or AMR Grids	Data Intensive
Accelerator Science		X	X	X	X	X	
Astrophysics	X	X	X	X	X	X	X
Chemistry	X	X	X	X			X
Climate			X		X	X	X
Combustion					X	X	X
Fusion	X	X		X	X	X	X
Lattice Gauge		X	X	X	X		
Material Science	X		X	X	X		

Antypas, Shalf, and Wasserman, **"NERSC-6 Workload Analysis and Benchmark Selection Process," LBNL report LBNL-72755.**

Machine Requirements

Algorithm Science areas	Dense linear algebra	Sparse linear algebra	Spectral Methods (FFT)s	Particle Methods	Structured Grids	Unstructured or AMR Grids	Data Intensive
Accelerator Science	High Flop/s rate	High performance memory system	High bisection bandwidth	High performance memory system	High flop/s rate	Low latency, efficient gather /scatter	Storage, Network Infrastructure
Astrophysics							
Chemistry							
Climate							
Combustion							
Fusion							
Lattice Gauge							
Material Science							

NERSC users require a system which performs adequately in all areas

Application Benchmarks

Benchmark	Science Area	Algorithm Space	Base Case Concurrency	Problem Description	Lang	Libraries
CAM	Climate (BER)	Navier Stokes CFD	56, 240 Strong scaling	D Grid, (~.5° resolution); 240 timesteps	F90	netCDF
GAMESS	Quantum Chem (BES)	Dense linear algebra	384, 1024 (Same as Ti-09)	DFT gradient, MP2 gradient	F77	DDI, BLAS
GTC	Fusion (FES)	PIC, finite difference	512, 2048 Weak scaling	100 particles per cell	F90	
IMPACT-T	Accelerator Physics (HEP)	PIC, FFT component	256,1024 Strong scaling	50 particles per cell	F90	
MAESTRO	Astrophysics (HEP)	Low Mach Hydro; block structured -grid multiphysics	512, 2048 Weak scaling	16 32^3 boxes per proc; 10 timesteps	F90	Boxlib
MILC	Lattice Gauge Physics (NP)	Conjugate gradient, sparse matrix; FFT	256, 1024, 8192 Weak scaling	8x8x8x9 Local Grid, ~70,000 iters	C, assem.	
PARATEC	Material Science (BES)	DFT; FFT, BLAS3	256, 1024 Strong scaling	686 Atoms, 1372 bands, 20 iters	F90	Scalapack, FFTW

Sustained System Performance (SSP)

- Aggregate, un-weighted measure of sustained computational capability relevant to NERSC's workload.
- Geometric Mean of the processing rates of seven applications multiplied by *N*, # of cores in the system.
  - Largest test cases used.
- Uses floating-point operation count predetermined on a reference system by NERSC.

$$\text{SSP in TFLOPS} = \frac{N * \sqrt[7]{\prod_i P_i}}{1000}$$

SSP Example

	Reference		Results	
Code	Tasks	GFLOP Count	Time	Rate per Core
cam	**240**	**57,669**	**408**	0.59
gamess	**1024**	**1,183,900**	**2478**	0.47
gtc	**2048**	**3,639,479**	**1493**	1.19
ImpactT	**1024**	**399,414**	**627**	0.62
maestro	**2048**	**1,122,394**	**2570**	0.21
milc	**8192**	**7,337,756**	**1269**	0.71
paratec	**1024**	**1,206,376**	**540**	2.18
SSP for 19,344 cores				**13.1**

Rate Per Core = GFLOP count / (Tasks * Time)

Flop count measured on reference system

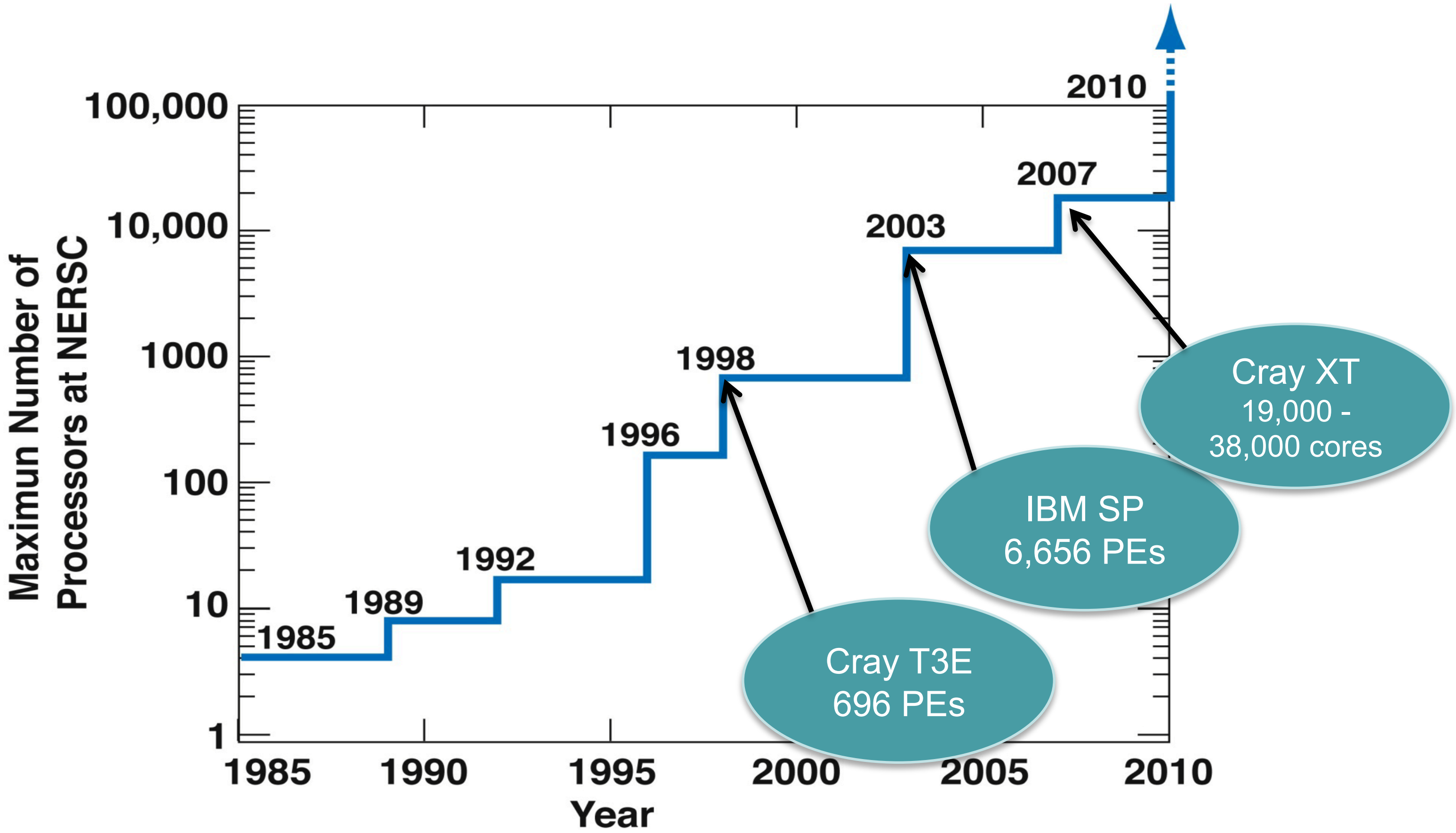
Measured wall clock time on system of interest

NERSC Next-Generation System

- **NERSC-6 (2010):**
  - **70-100 TF SSP goal**
  - **Today: 13 TF SSP on NERSC-5 (Franklin, ~20,000 cores)**

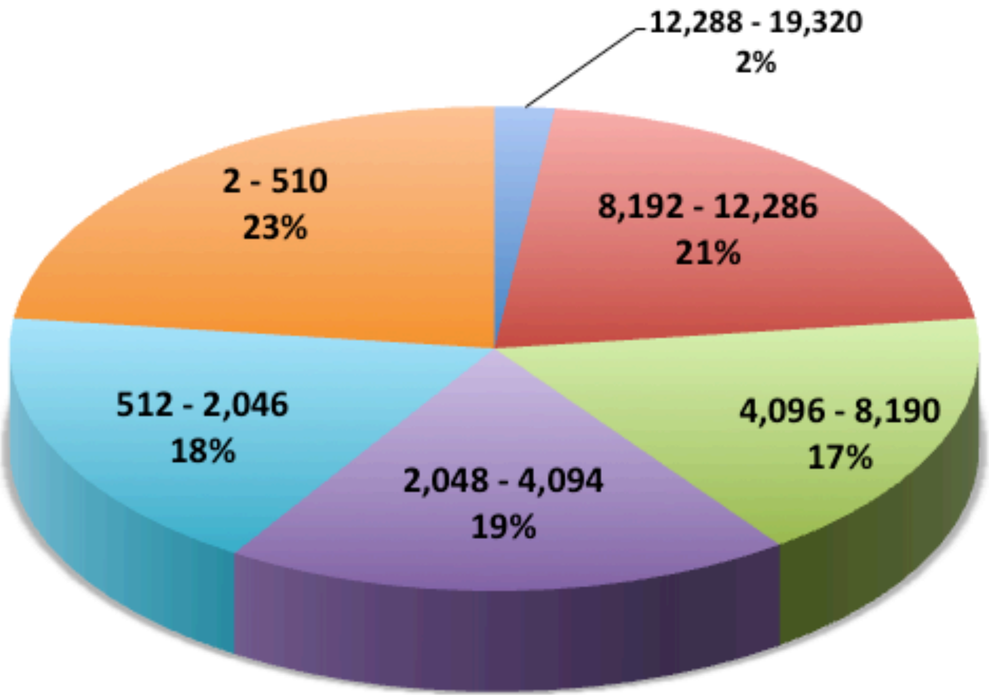
    => ~100,000-core NERSC-6

Parallelism at NERSC: Historical



By 2011 NERSC will run a system with about 100K cores in production mode for its 2000+ user base.

Parallelism at NERSC Today



Raw Hours used on Franklin FY08 Q1-Q3 by # of cores (Raw Hours = wallclock hours * nodes * 2 CPUs/node)

- **Parallelism levels are reasonable for this point in time. But why might this have to change?**

Concurrency Level is Constrained by System Size

New Architecture Constraints

- 15 years of exponential growth in processor rate has ended.

- Moore's Law is alive and well.
  - But industry response is to double number of cores per socket every ~18 months

- Memory Capacity is Not Growing at Same Rate as Transistors / Cores => Less Memory / Core

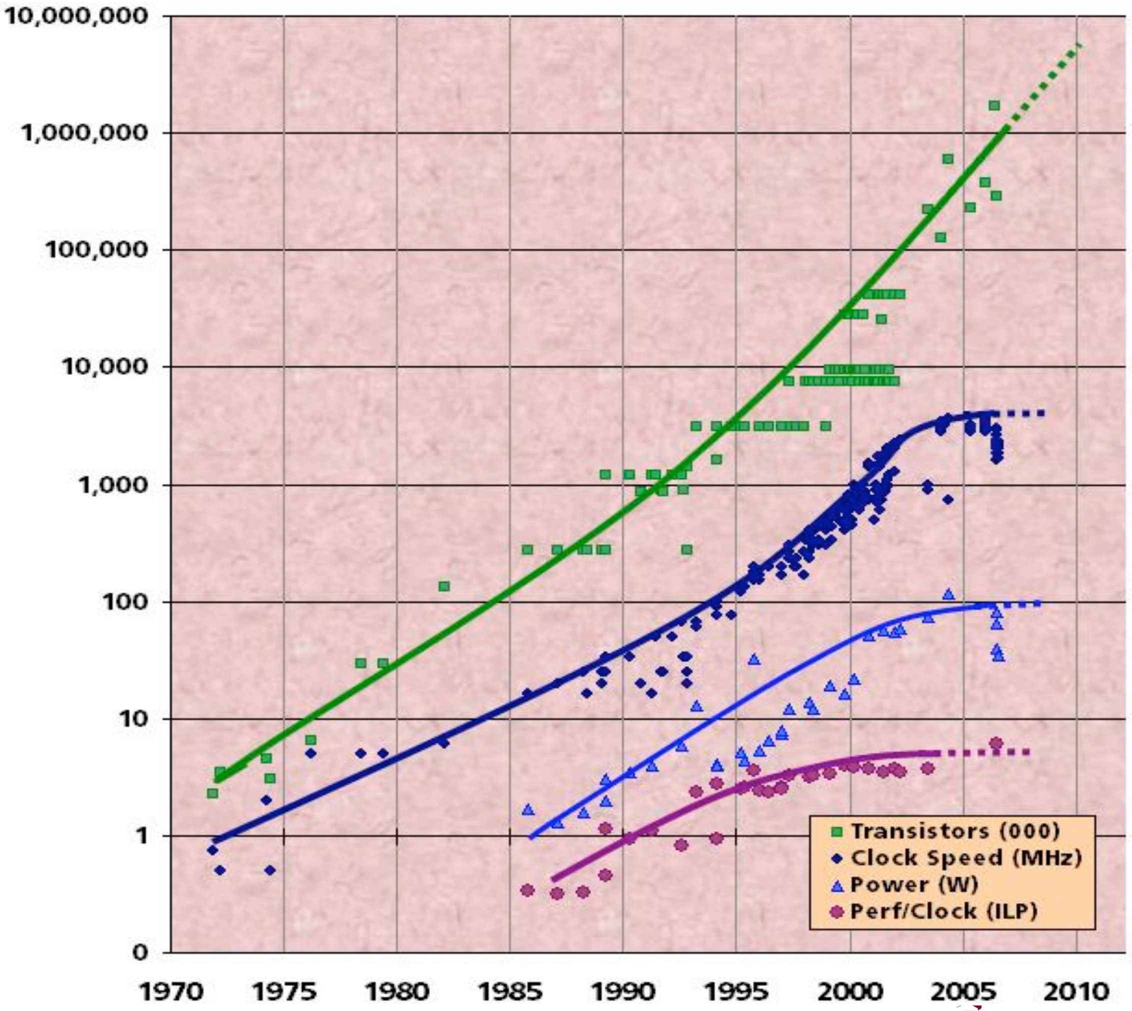
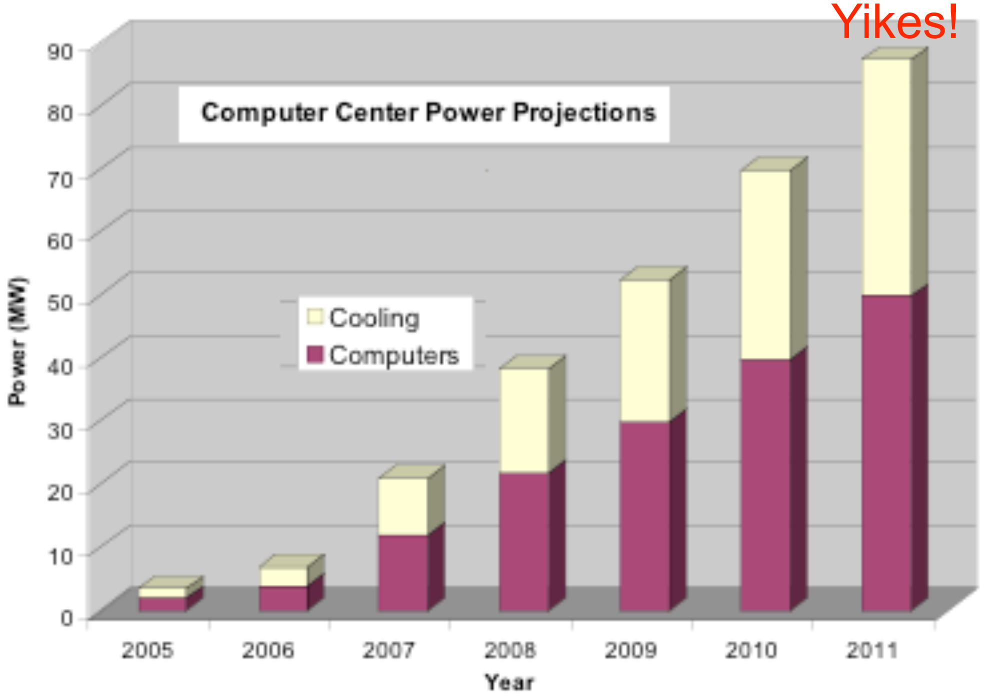


Figure courtesy of Kunle Olukotun, Lance Hammond, Herb Sutter, and Burton Smith

Primary Hardware Problems

- Power limits leading-edge chip & system designs
  - ASC "Sequoia" system budget = 15 MW/year
  - ORNL $33M/year projected power+cooling costs in 2010
- Yield on leading edge processes dropping dramatically
  - IBM quotes yields of 10 – 20% on 8-processor Cell

Yikes!



Cost estimates based on $0.05 kW/hr

- Verification for leading edge chips is becoming unmanageable.
  - Verification teams > design teams on leading edge processors

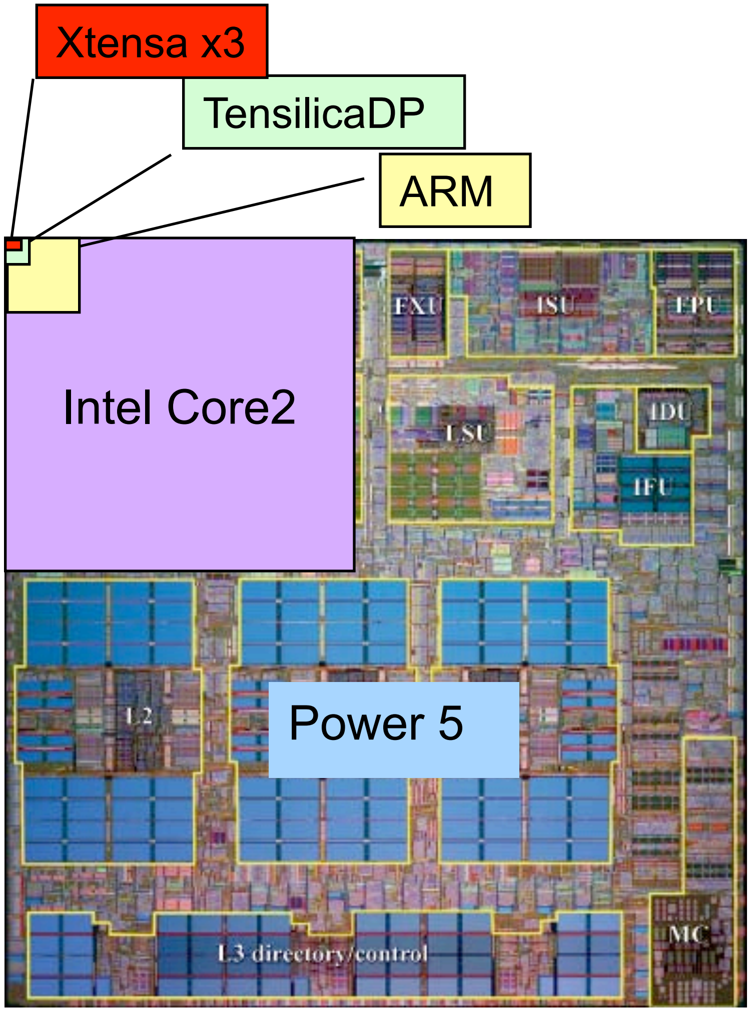
Hardware Constraints Lead from Multicore…

- **Multicore: current trajectory**
  - Stay with current fastest core design
  - Replicate every 18 months (2, 4, 8 . . .etc.)
  - Advantage: Do not alienate serial workload
  - Example: Intel Core2 Duo (2 cores), Tigerton (4), Nehalem (8); Intel Madison (2), Tukwila (4); AMD Barcelona (4 cores), Shanghai (4), Istanbul (6), …;
  - Big and still relatively power hungry

… to Manycore

- **Manycore: small is beautiful!**
  - Simplify cores (shorter pipelines, lower clock frequencies, in-order processing + SIMD processing)
  - Redundant processor advantage: easier verification, defect tolerance, highest compute/surface-area, best power efficiency
    - Not much slower than large cores
  - Examples: Cell SPE (8 cores), Nvidia G80 (128 cores), Intel Polaris (80 cores), Cisco/Tensilica Metro (188 cores), Sun Niagara/2
  - What about semi-embedded (BG)? Converging in this direction?
  - Hedge: Heterogenous Multicore

How Small is “Small”



- **Power5 (Server)**
  - **389mm^2**
  - **120W@1900MHz**
- **Intel Core2 sc (laptop)**
  - **130mm^2**
  - **15W@1000MHz**
- **ARM Cortex A8 (automobiles)**
  - **5mm^2**
  - **0.8W@800MHz**
- **Tensilica DP (cell phones / printers)**
  - **0.8mm^2**
  - **0.09W@600MHz**
- **Tensilica Xtensa (Cisco router)**
  - **0.32mm^2 for 3!**
  - **0.05W@600MHz**

Slide courtesy of John Shalf (LBNL)

New cores operate at 1/3 - 1/10th efficiency of largest chip, but take up 1/100 space and consume 1/20 the power

Statement of the Problem

- Oldest CW: Innovation trickles down from High End Computing to mainframes.
- Older CW: Innovation in processor design for PCs (COTS) trickles <u>up</u> to High End Computing
- New CW: World revolves around consumer devices.
  - Better at computational/power efficiency
  - Better at cost-effectiveness
  - Examples:
    - Motorola Razor Cell Phone already has 8 cores
    - Cisco CRS-1 router has 188 Tensilica cores
  - <u>Not</u> the same as COTS
- (HPC hasn’t been in the driver’s seat since ~1962.)

Industry's Problem

- Parallelism is the primary path forward (unless you're content with 2008 application speed).

- Shift to Multicore / Manycore is happening without consensus on a parallel programming model.

Source: "The Landscape of Parallel Computing Research: A View From Berkeley," **http://view.eecs.berkeley.edu/**

More than any time in history, mankind faces a crossroads. One path leads to despair and utter hopelessness, the other to total extinction.

Let us pray that we have the wisdom

to choose correctly.

- Woody Allen

What Does it Mean for NERSC?

- Need to support existing production user base.
- Immediate need to select best future machine.
  - Anticipate some bids with “accelerators” for NERSC-6
    - Benchmarking must adapt.
  - New emphasis on power efficiency
    - 3.5 MW power limit for Oakland Scientific Facility (OSF)
    - Require 480VAC 3-phase power distribution for efficiency
  - Evaluate improved cooling efficiency if systems operate at high-end of ASHRAE allowable thermal range
  - Memory limitations - Increasing source of power consumption
  - Expect bids with constrained memory
    - Benchmarking must adapt.

NERSC Short-Term Response

- Two benchmarking modes for NERSC-6:
  - Base case: MPI-only, fixed concurrency, no code changes
    - Concurrency change for constrained memory allowed
  - Optimized case: more (or fewer) cores, OpenMP, code modifications, accelerators, any/all of the above
    - “Full Fury” mode

What Does it Mean for NERSC?

- Longer-term: Can we program multicore / manycore?
  - 2 cores for video, 1 for MS Word, 1 for browser, 76 for virus / spam check? *
  - Optimizing performance-per-watt necessarily includes consideration of programmability.
- Opportunity: Leverage local research in
  - Algorithms: efficiency & unprecedented parallelism
  - Programming models / languages
  - Tuning methods
  - Architecture

*Source: J. Kubiatowicz, 2-day short course on parallel computing," **http://parlab.eecs.berkeley.edu**

What Does it Mean for NERSC?

- Longer-term: Can we program multicore / manycore?
  - 2 cores for video, 1 for MS Word, 1 for browser, 76 for virus / spam check? *
  - Optimizing performance-per-watt necessarily includes consideration of programmability.
- Opportunity: Leverage local research in
  - Algorithms: efficiency & unprecedented parallelism
  - Programming models / languages
  - Tuning methods
  - Architecture

Algorithmic Trends

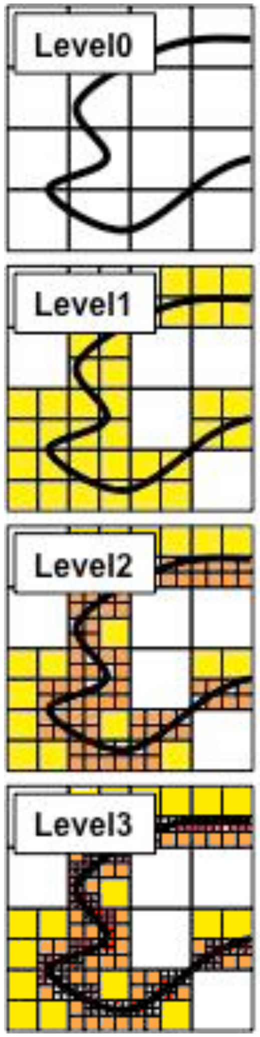
- HPC thrived on weak scaling for past ~15 years.
- Flat CPU performance increases emphasis on strong scaling.
  - Ability to accommodate Moore's Law increase in concurrency.
  - Partially due to increasing memory limitations.
  - Results in small inter-processor messages, greater latency dependence
- Timestepping increasingly driven towards implicit or semi-implicit stepping schemes
  - Requires support for fast global reductions
- Spatially adaptive approaches (AMR)

NERSC Short-Term Response

- Include benchmarks representing forward -looking algorithms/languages.
  - Adaptive Mesh Refinement (AMR) proxy
  - Implicit methods
  - UPC

AMR Performance Challenges

- AMR offers substantial benefits over fully-explicit uniform grid methods
  - Especially in reduced memory environments
- Problems:
  - non-uniform memory access,
  - extra metadata / grid bookkeeping,
  - irregular inter-processor communication,
  - Methodology for performance measurement.



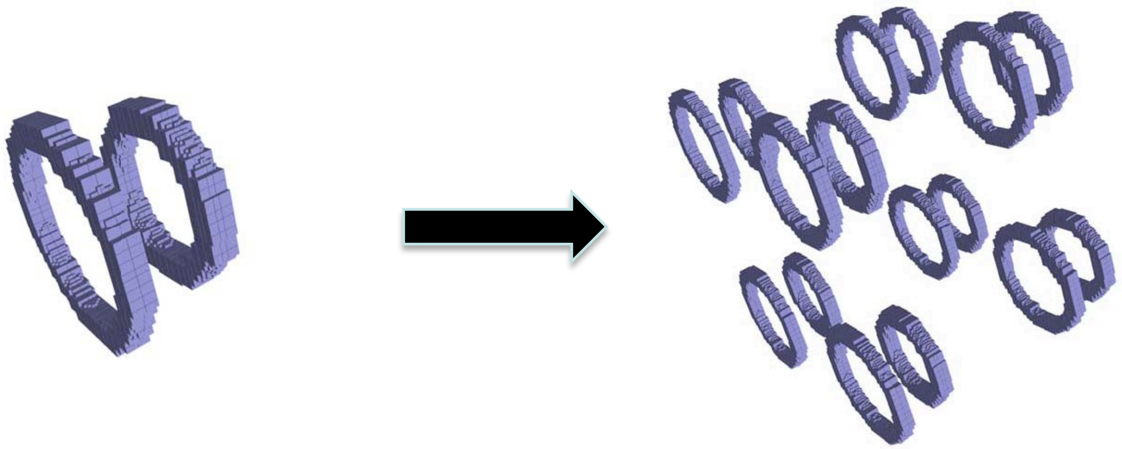
AMR Performance Challenges

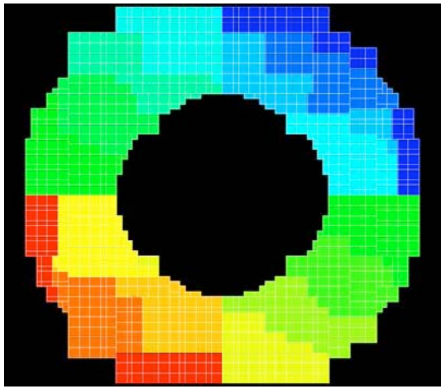
- Problem: how to weak-scale AMR
  - Could scale coarsest grid but then adaptivity doesn't match.
- Solution: Take a single grid hierarchy and scale by making identical copies.
  - Work/memory per core remains constant

P. Colella, J. Bell, N. Keen, T. Ligocki, M. Lijewski, and B. van Straalen, "Performance and Scaling of Locally-Structured Grid Methods for PDEs," J. Phys: Conf. Series **78** (2007) 012013

NERSC/LBNL AMR Benchmark

- "Stripped-down" Poisson solver
- C++ Code, scales to 8192 cores
- Very sensitive to OS "jitter"



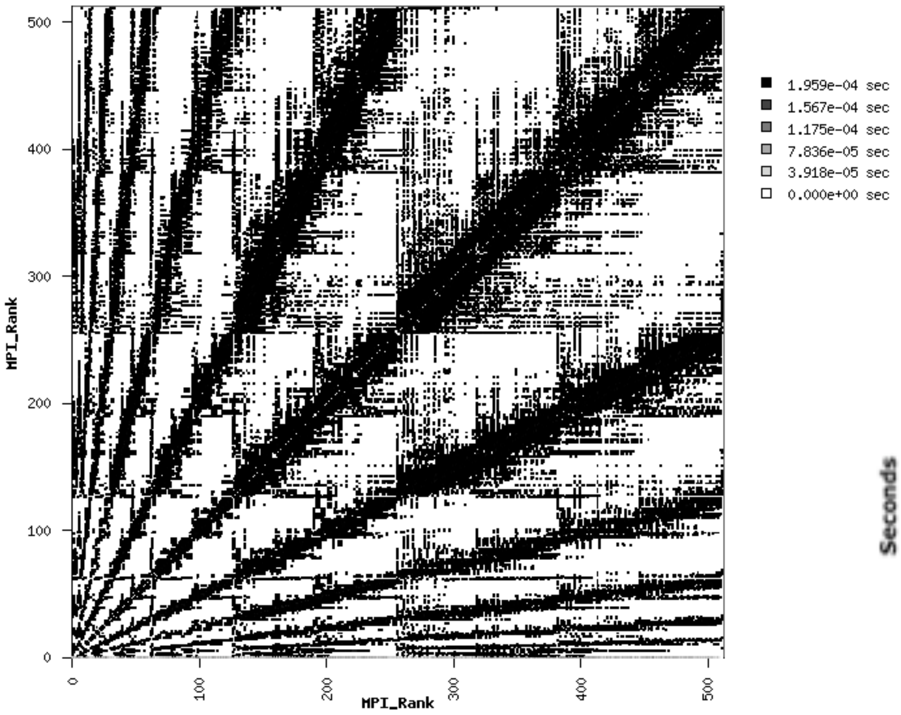


AMR Architectural Stress Points

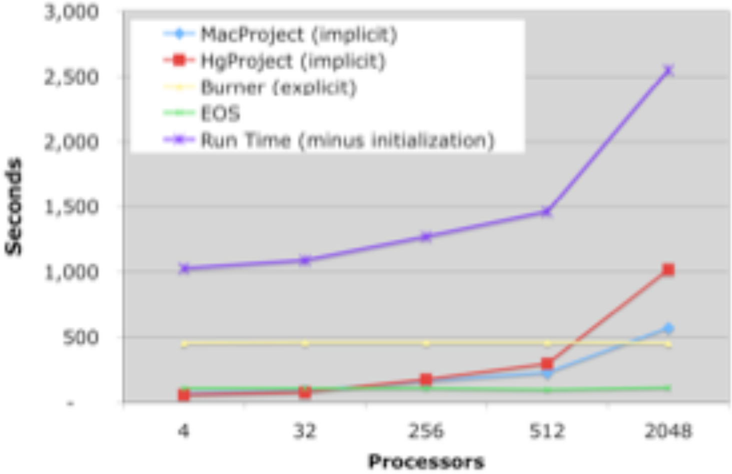
- NERSC “Maestro” benchmark code
- Low Mach number flow
- Represents both combustion and Supernova explosion science.
- AMR overhead reflected in low computational intensity (0.24 FLOPs per memory ref.)
- “Unusual” communication topology:

MAESTRO Communication

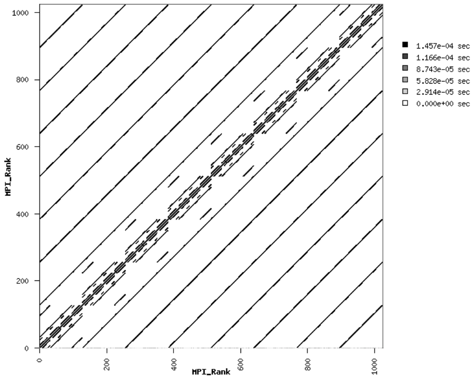
- Nearest neighbor topology measured using NERSC IPM tool (http://ipm-hpc.sourceforge.net/)



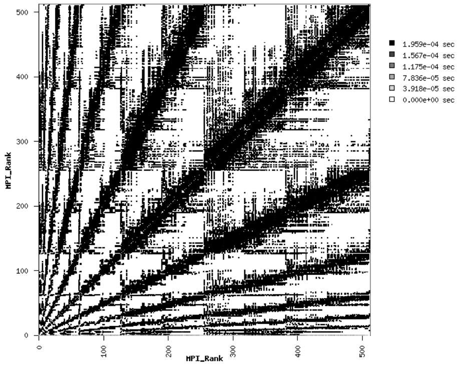
- Clumping effect results from load balancing



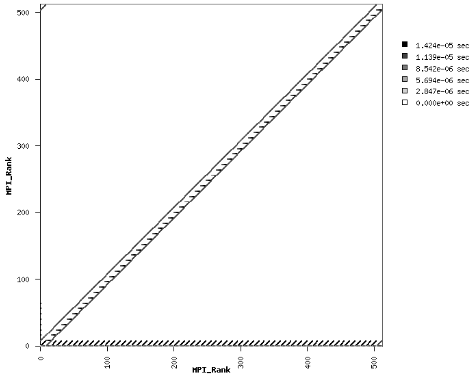
Comm Topo for Other Codes



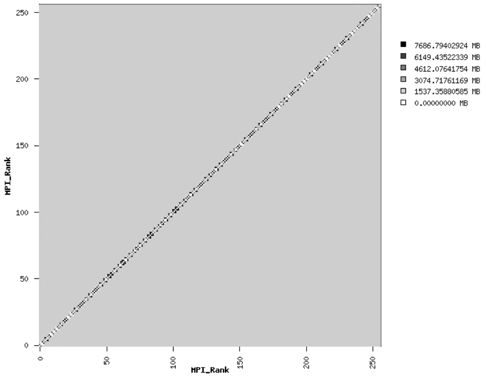
MILC



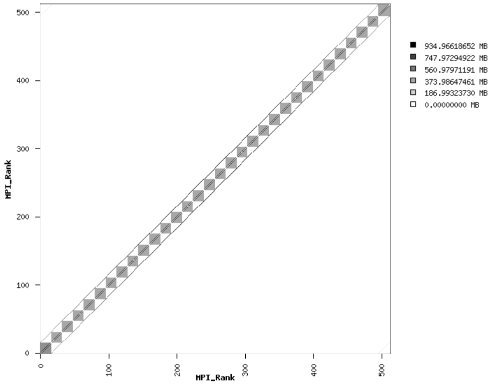
MAESTRO



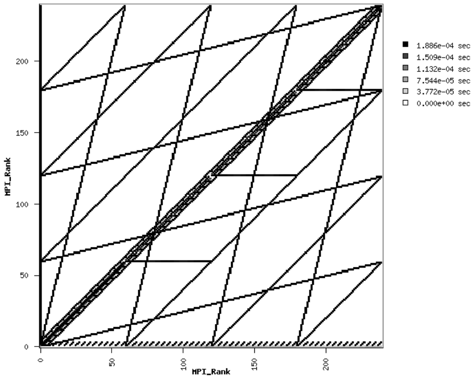
GTC



PARATEC



IMPACT-T



CAM

Applications are topology sensitive and interconnect hierarchy is deepening.

What Does it Mean for NERSC?

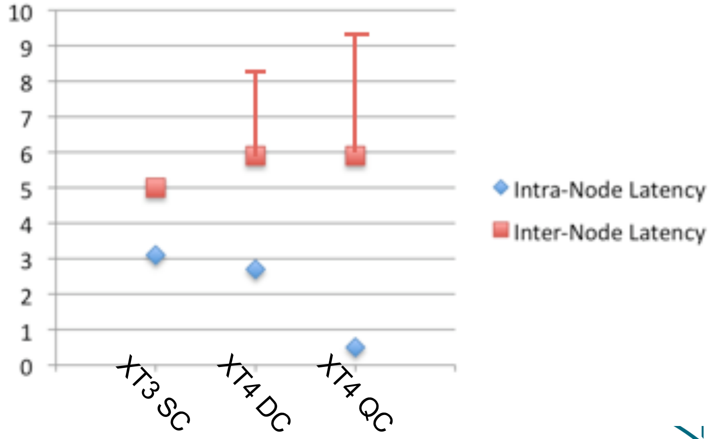
- Longer-term: Can we program multicore / manycore?
  - 2 cores for video, 1 for MS Word, 1 for browser, 76 for virus / spam check? *
  - Optimizing performance-per-watt necessarily includes consideration of programmability.
- Opportunity: Leverage local research in
  - Algorithms: efficiency & unprecedented parallelism
  - Programming models / languages
  - Tuning methods
  - Architecture

Multicore is Not a Familiar Programming Target

- What about Message Passing on a chip?
  - Path of least resistance will work for a while
    - Apps port easily; requires modest infrastructure work (multicore-optimized MPI)
  - But MPI buffers / data structures grow as O(N) or O(N^2): a problem for constrained memory (reduces weak scaling efficacy)
  - Message traffic overwhelms NIC in some cases
  - Requires lighter-weight messaging (weak point of MPI)

Multicore is NOT a Familiar Programming Target

- What about SMP on a chip?
  - Hybrid Model (MPI+OpenMP): Obvious next step but long history with only limited success.
    - People don't want two programming models.
    - Very difficult to debug
  - Manycore/Multicore is NOT an SMP on a chip
    - 10-100x higher bandwidth on chip
    - 10-100x lower latency on chip
  - SMP model ignores potential for much tighter coupling of cores



Multicore is NOT a Familiar Programming Target

- What about hybrid MPI + ???
  - – LANL Roadrunner experiment
  - – CEA Bull system with Intel Nehalem + GPGPUs.
  - – Intel, Microsoft, Apple efforts: useful for scientific programming?
  - – PeakStream (aka Google), RapidMind, …
- Auto-parallelization will not work
  - – But auto-tuning might.

NERSC FFT UPC Benchmark

- NAS Parallel Benchmark FT Class D
- Coded in UPC by K. Yelick and grad students
  - Uses pthreads
- Commercial compilers available on Cray, SGI, HP
- Proxy for one-sided communication and overlap methods – applicable to chemistry applications and others.

C. Bell, D. Bonachaea, R. Nishtala, K. Yelick, “Optimizing Bandwidth Limited Problems Using One-Sided Communication and Overlap,” IPDPS2006.
http://upc.lbl.gov/publications/upc_bisection_IPDPS06.pdf

What Does it Mean for NERSC?

- Longer-term: Can we program multicore / manycore?
  - 2 cores for video, 1 for MS Word, 1 for browser, 76 for virus / spam check? *
  - Optimizing performance-per-watt necessarily includes consideration of programmability.
- Opportunity: Leverage local research in
  - Algorithms: efficiency & unprecedented parallelism
  - Programming models / languages
  - Tuning methods
  - Architecture

Programmability

- UC Berkeley two-layer approach to:
  - Efficiency Layer (10% of today's programmers)
    - Expert programmers build Frameworks & Libraries …
    - "Bare metal" efficiency but hide it from …
  - Productivity Layer (90% of today's programmers)
    - Domain experts build parallel apps using frameworks & libraries
- Leverage efforts in frameworks/community codes, e.g., Chombo, Cactus, SIERRA, UPIC, CCA, EMSF, Overture, SAMRAI)
  - Hide complexity using good software engineering

Autotuning Research @ LBNL

(and elsewhere, e.g., Dongarra)

- Sacrifice up-front machine time for continued reuse of auto-optimized kernel on range of architectures.
- Automates search over possible implementations
- Auto-tune by heuristics or exhaustive search
  - Perl script generates many versions
  - Autotuner analyzes/runs kernels
  - In-core (ILP, SIMD, unroll, …)
  - Memory latency (prefetch, reorder loops, …)
  - Cache (blocking, …)
  - Parallel multi-socket, multi-core via threads
  - Including NUMA

Compilers with maximum optimization are not delivering scalable performance

LBNL Autotuning References

S. Williams, J. Carter, L. Oliker, J. Shalf, K. Yelick, "Lattice Boltzmann Simulation Optimization on Leading Multicore Platforms", International Parallel & Distributed Processing Symposium (IPDPS), 2008. Best Paper, Application Track

K.Datta, M.Murphy, V. Volkov, S. Williams, J. Carter, L. Oliker, D. Patterson, J. Shalf, K. Yelick, “Stencil Computation Optimization and Autotuning on State-of-the-Art Multicore Architectures”, SC08 (to appear), 2008 *(in press)*.

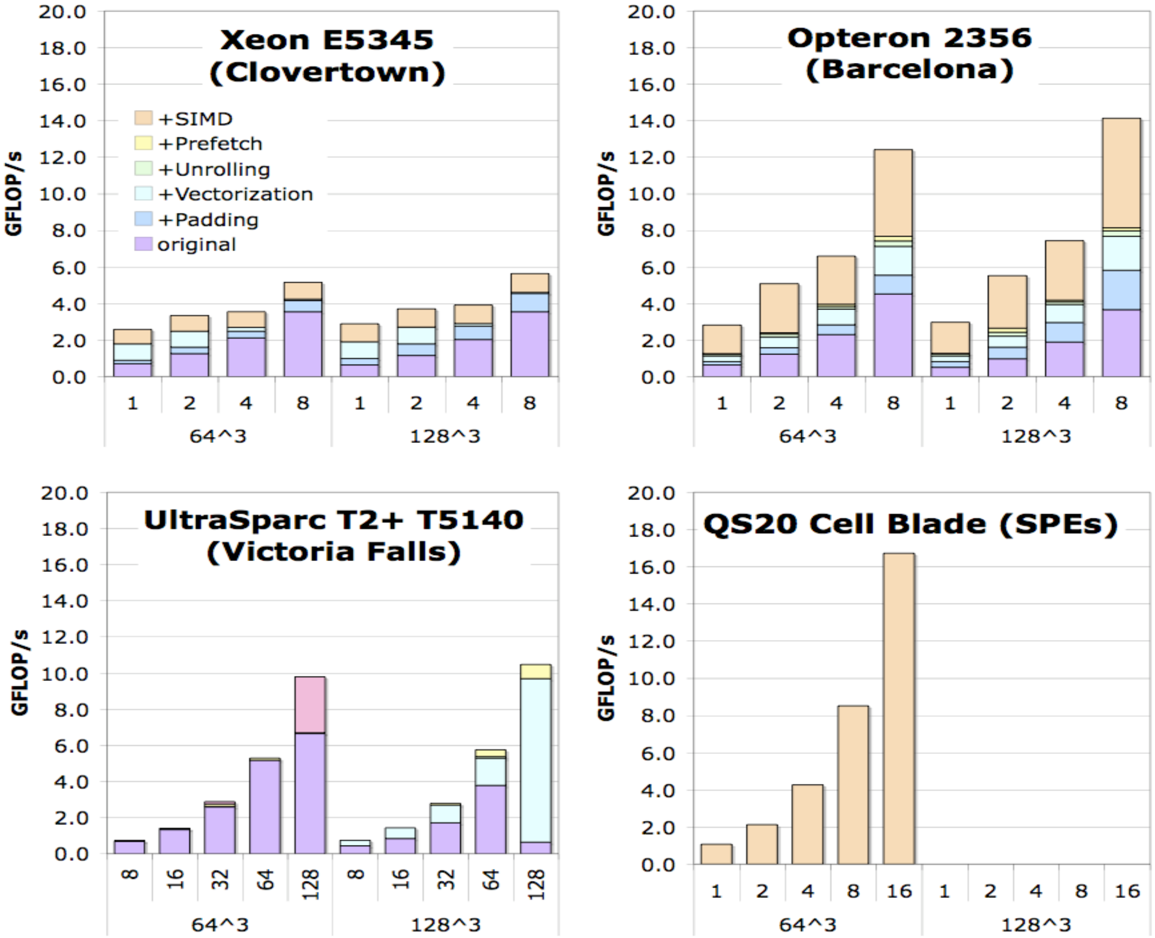
S. Williams, L. Oliker, R. Vuduc, J. Shalf, K. Yelick, J. Demmel, "*Optimization of Sparse Matrix-Vector Multiplication on Emerging Multicore Platforms*", SC07.

K. Datta, S. Kamil, S. Williams, L. Oliker, J. Shalf, K. Yelick, "*Optimization and Performance Modeling of Stencil Computations on Modern Microprocessors", SIAM Review, 2008 (in press).*

R. Vuduc, J. Demmel, K. Yelick, OSKI, http://bebop.cs.berkeley.edu/oski/

Lattice-Boltzman Performance

(auto-tuned)



- Auto-tuning avoids cache conflict and TLB capacity misses
- Exploits SIMD where the compiler doesn't
- Include a SPE/Local Store optimized version
- Performance approaches maximum provided by architecture
- Tuning approach is highly architecture dependent.

■ +small pages

■ +Explicit SIMDization

■ +SW Prefetching

■ +Unrolling

■ +Vectorization

■ +Padding

■ Naïve+NUMA

S. Williams, J. Carter, L. Oliker, J. Shalf, K. Yelick, "Lattice Boltzmann Simulation Optimization on Leading Multicore Platforms", International Parallel & Distributed Processing Symposium (IPDPS), 2008. Best Paper, Application Track

*collision() only

What Does it Mean for NERSC?

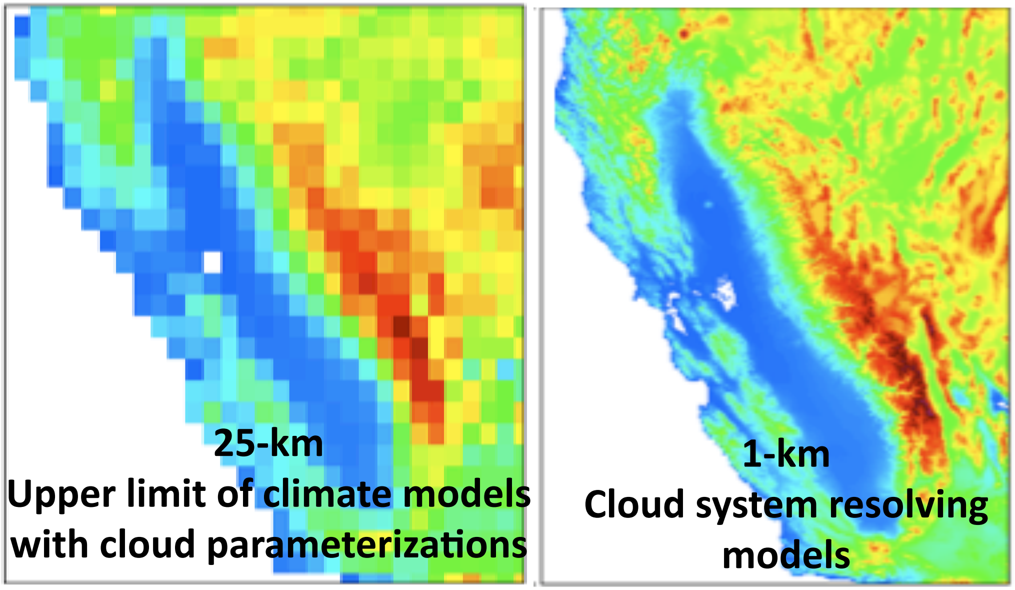
- Longer-term: Can we program multicore / manycore?
  - 2 cores for video, 1 for MS Word, 1 for browser, 76 for virus / spam check? *
  - Optimizing performance-per-watt necessarily includes consideration of programmability.
- Opportunity: Leverage local research in
  - Algorithms: efficiency & unprecedented parallelism
  - Programming models / languages
  - Tuning methods
  - Architecture

Green Flash Overview

- Explore energy-efficient computing, from system design to apps
- Research effort: study feasibility, share insight with community
- Elements of the approach:
  - Choose the science target first (climate initially)
  - Design systems for the application (rather than the reverse)
  - Evolve HW & SW together using hardware emulation and auto -tuning
- What is new about this approach:
  - Leverage commodity processes used to design power efficient embedded devices.
  - Auto-tuning to automate mapping of algorithm to complex hardware
  - RAMP: Fast FPGA-accelerated emulation of new chip designs
- Applicable to broad range of scientific computing applications?

Motivation: 1-km Climate Models

- Direct simulation of cloud systems replacing current statistical parameterization.
- 1000x real time simulation speed.
- Estimate 10 PF sustained per simulation (~200 PF peak)
- Simultaneous algorithm development w/ NERSC.



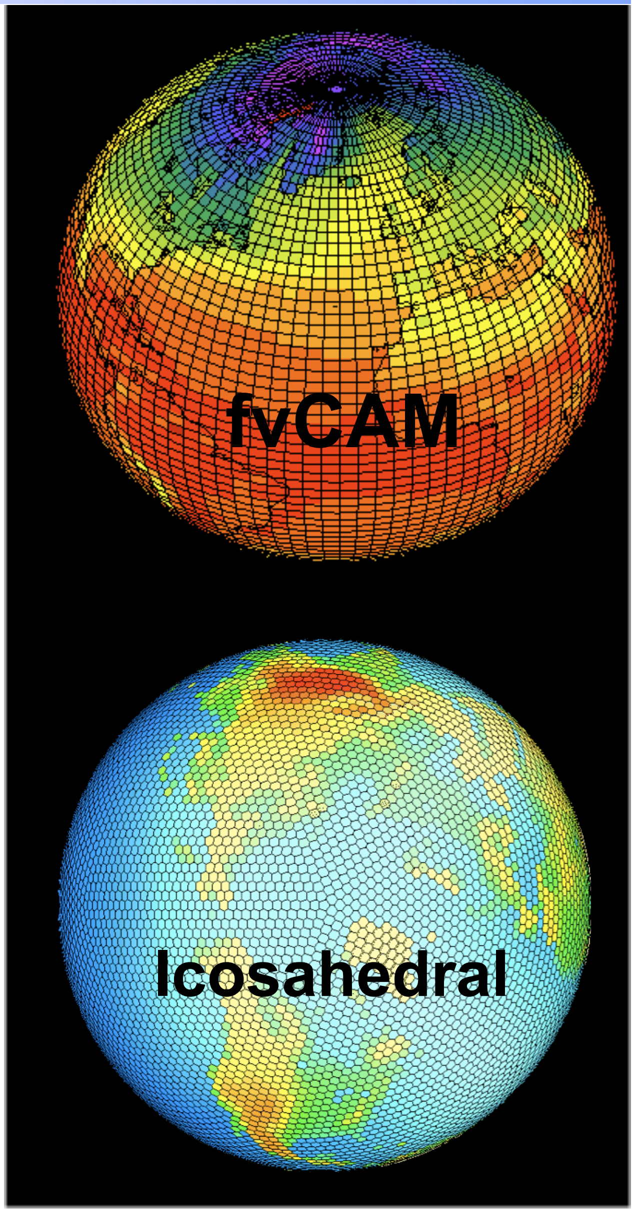
25-km
Upper limit of climate models with cloud parameterizations

1-km
Cloud system resolving models

M. Wehner, L. Oliker, and J. Shalf, "**Towards Ultra-High Resolution Models of Climate and Weather,"** Int. J. High Perf. Comp. App, **May 2008, 22, No. 2**

Algorithm Assumptions

- **Based on CAM performance model.**
- **Existing lat.-long. based advection algorithm breaks down before 1-km scale**
  - **Grid cell aspect ratio at pole is ~10000**
  - **Advection time step is problematic.**
- **Ultimately requires new discretization**
  - **Must expose sufficient parallelism to exploit power-efficient design**
  - **Prof. D. Randall, Colorado St. U., use Icosahedral grid, special INCITE grant.**
  - **Uniform cell aspect ratio across globe**
  - **Scales to ~20B cells, 20M subdomains**



Green Flash Strawman Design

- **Examined three different approaches (in 2008 technology); collaboration with Tensilica®**
  - **Compare Opteron "commodity" (890K sockets), BG/L "generic embedded" (1.8M) and Tensilica "custom embedded" (116K sockets, 3.7M cores@650MHz)**
  - **Result: $75, 3MW, 10PF sustained using 2008 lithography and climate-custom core**
    - **Approach uses commodity design tools; not the same as full -custom design**

M. Wehner, L. Oliker, and J. Shalf, "**Towards Ultra-High Resolution Models of Climate and Weather,"** Int. J. High Perf. Comp. App, **May 2008, 22, No. 2**

http://www.lbl.gov/CS/html/greenflash.html

What Does it Mean for NERSC?

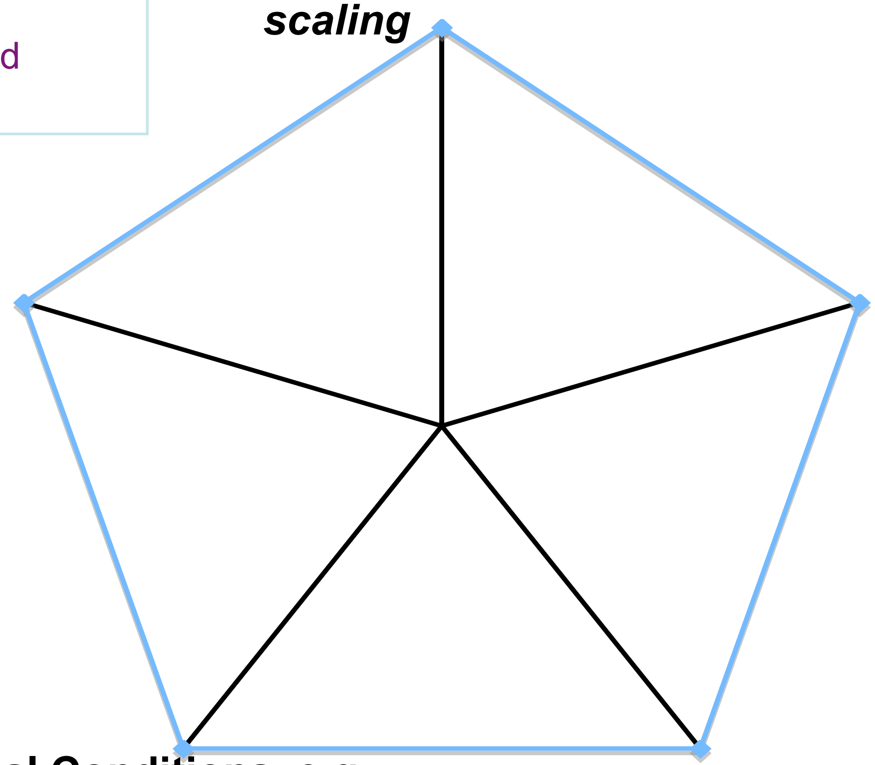
- Need to support existing production user base.
- Need to select best future machine. Benchmarking must adapt.
- Can we program multicore / manycore?
  - 2 cores for video, 1 for MS Word, 1 for browser, 76 for virus/ spam check?
  - Optimizing performance-per-watt necessarily includes consideration of programmability.
- Leverage local research in
  - Algorithms
  - Programming models / languages
  - Tuning methods
  - Architecture

Other Concerns Not Addressed

- OS issues
- I/O
- Hardware/SW Transactional memory
- Fault tolerant software
- Debugging / program correctness

Scaling Computational Science

Inspired by P. Kent, “Computational Challenges in Nanoscience: an *ab initio* Perspective”, Peta08 workshop, Hawaii (2008) and Jonathan Carter (NERSC).



Length, Spatial extent, #Atoms, *Weak scaling*

Time scale Optimizations, *Strong scaling*

Simulation method, e.g. DFT or CC, LES or DNS

Initial Conditions, e.g. molecule, boundaries, *Ensembles*

Convergence, systematic errors due to cutoffs, within one method

Scientists Need More Than FLOP/s

- Performance — How fast will a code run?
- Effectiveness — How many codes can a system process?
- Reliability — How often is the system available and operating correctly?
- Consistency — How often will the system process users' work as fast as it can?
- Usability — How easy is it for users to get the system to go as fast as possible?

PERCU: NERSC's method for ensuring scientific computing success.

Acknowledgements

- A large number of individuals have contributed to energy efficiency in computing at the Lab and to this presentation, including:

- Katie Anytpas (NERSC), David Bailey (CRD), Shoaib Kamil (CRD), Lenny Oliker (CRD), John Shalf (NERSC), Erich Strohmaier (CRD), Michael Wehner Kathy Yelick (NERSC/CRD), Horst Simon (CS), Jonathan Carter (NERSC)

THANK YOU.