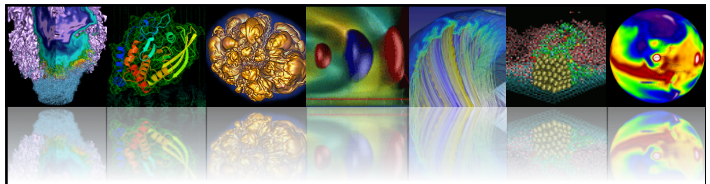


Big Data, Big Iron and the Future of HPC

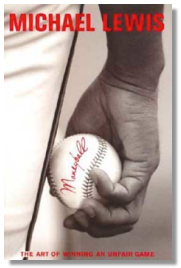
Kathy Yelick

Associate Laboratory Director of Computing Sciences

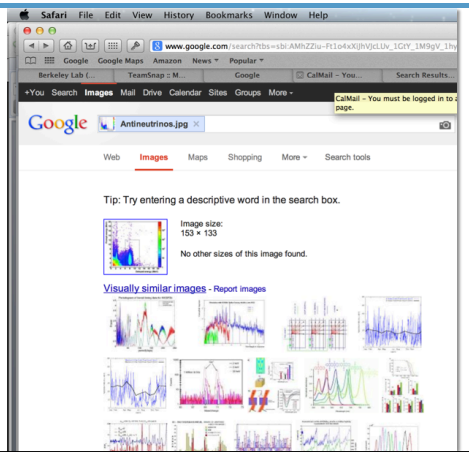
Lawrence Berkeley National Laboratory

EECS Professor, UC Berkeley

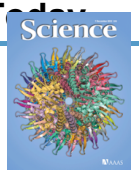
"Big Data" Changes Everything…What about Science?

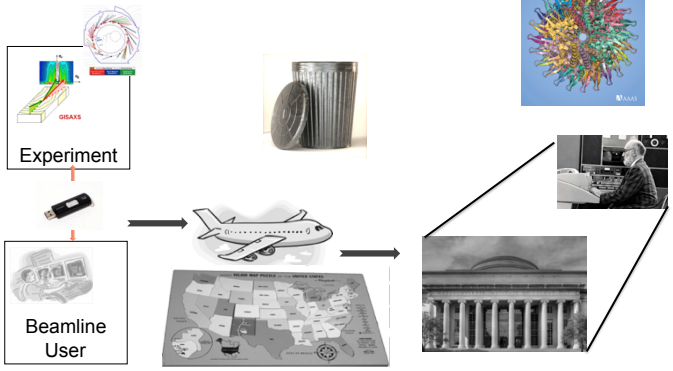


Transforming Science: Finding Data

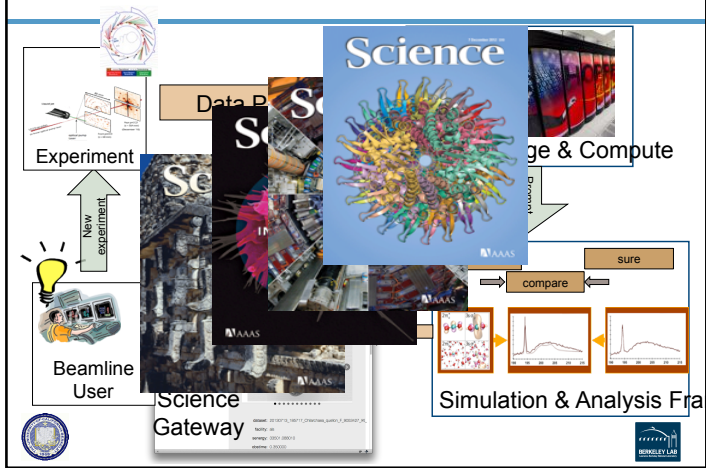


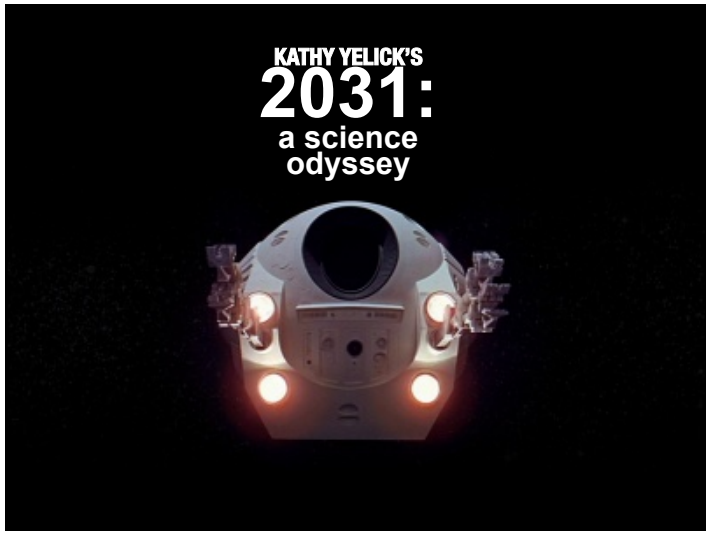
Scientific Workflow Today





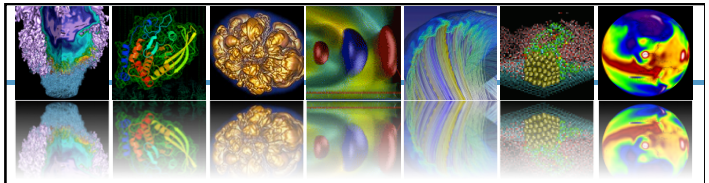
Scientific Workflow envisioned





Life of a Scientist in 2031

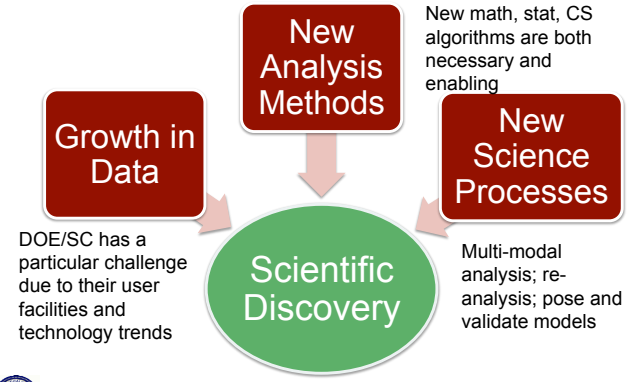
- No personal/departmental computers
- Users don't login to HPC Facilities
- Travel replaced by telepresence
- Lecturers teach millions of students
- Theorems proven by online communities
- Laboratory work is outsourced
- Experimental facilities are used remotely
- All scientific data is (eventually) open
- Big science and team science democratized



Extreme Data Science

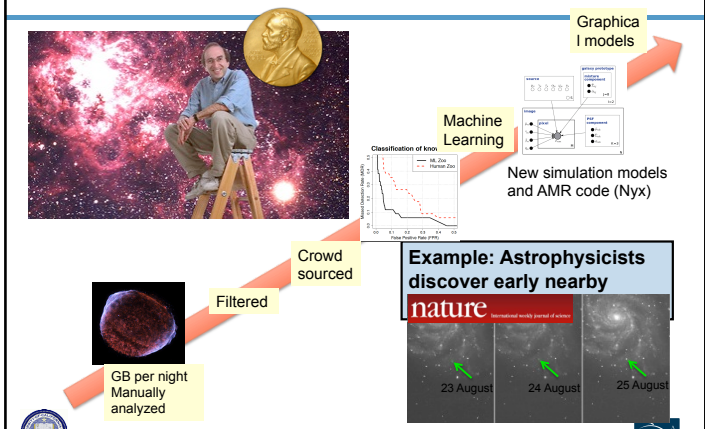
The scientific process is poised to undergo a radical transformation based on the ability to access, analyze, simulate and combine large and complex data sets.

Goal: To enable new modes of scientific discovery



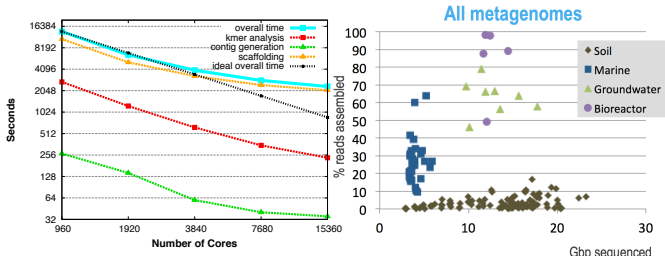
- 9 -

Data in Astrophysics: The Challenge is Systematics



Energy Genomics Grand Challenge

- Plant genome: former grand challenge to assemble wheat
- Next: metagenome assembly, currently limited by memory (and time)



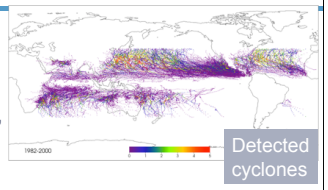
- SMP algorithms fail at ~100GB
- Throwing data away to process at all!
- Distributed memory assembly needed
- FPGA refactored code is better
- Still fails at ~850GB

Essential to understanding microbial dark-mater and their impacts

Filter and Pattern Match with Machine Learning

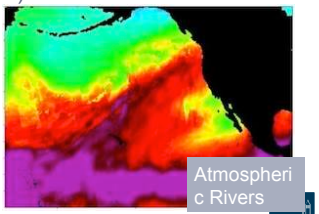
TECA Toolkit

- Automatic detection of cyclones, atmospheric rivers, and more
- Single data set is 100 TB
- Scalable analysis (80K cores): 9 years → 1 hour



Ongoing work

- Pattern detection using machine learning



Mantissa Project, Prabhat

12

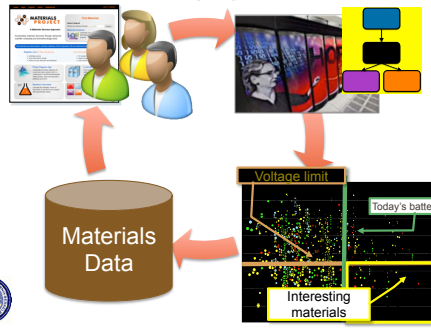
Filtering, De-Noise and Curating Data

AmeriFlux & FLUXNET: 750 users access carbon sensor data from 960 carbon flux data years

Arno Penzias and Robert Wilson discover Cosmic Microwave Background in 1965

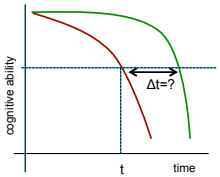
Re-Use and Re-Analyze Previously Collected Data

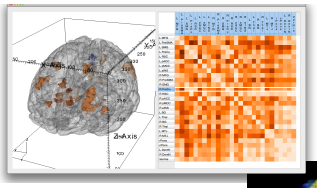
- **Materials Genome Initiative**
  - **Materials Project: 4500 users 18 months!**
  - **"World-Changing Idea of 2013"**

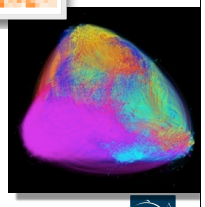


Brain Imaging: Multi-Modal Analysis and Data Fusion

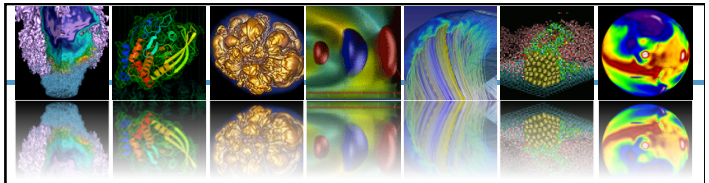
Analyze brain using multiple modalities and scales





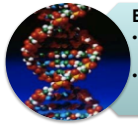


- **Detection of regions across community**
  - 100 individuals takes 18 days right now
- **Graph to classify disease**
  - Features: biomarkers, image modalities
  - Use hierarchy of regions from Pearson distance

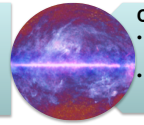


Science Data is Big (and Growing)

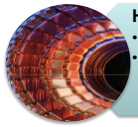
"Big Data" Challenges in Science
Volume, velocity, variety, and veracity



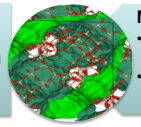
Biology
- *Volume:* Petabytes now; computation-limited
- *Variety*: multi-modal analysis on bioimages



Cosmology / Astronomy:
- *Volume:* 1000x increase every 15 years
- *Variety:* combine data sources for accuracy

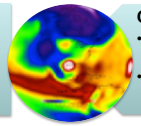


High Energy Physics
- *Volume*: 3-5x in 5 years
- *Velocity*: real-time filtering adapts to intended observation



Materials:
- *Variety:* multiple models and experimental data
- *Veracity:* quality and resolution of simulations

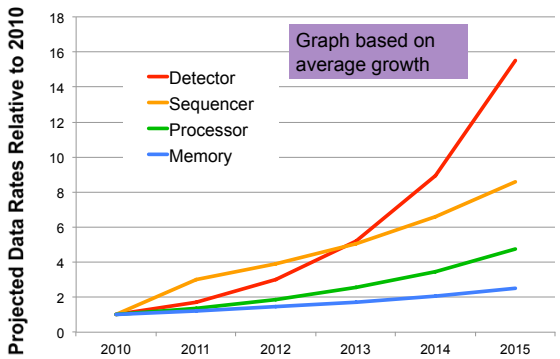
Light Sources
- *Velocity:* CCDs outpacing Moore's Law
- *Veracity:* noisy data for 3D reconstruction



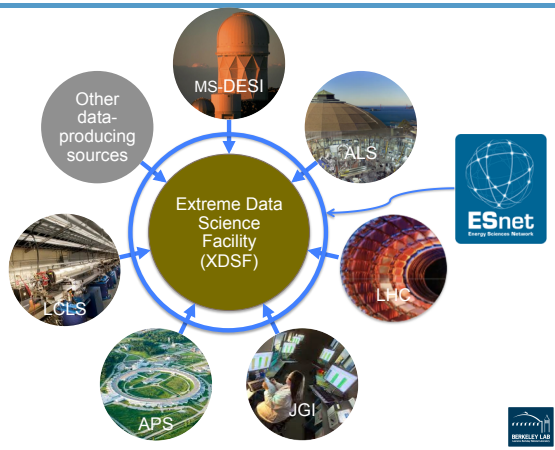
Climate
- *Volume:* Hundreds of exabytes by 2020
- *Veracity:* Reanalysis of 100-year-old sparse data

- 17 -

Data Growth is Outpacing Computing Growth

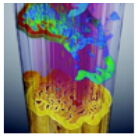


Superfacility Concept



Transform Experimental Science

Cosmology



Light sources

Genomics

Energy Technologies

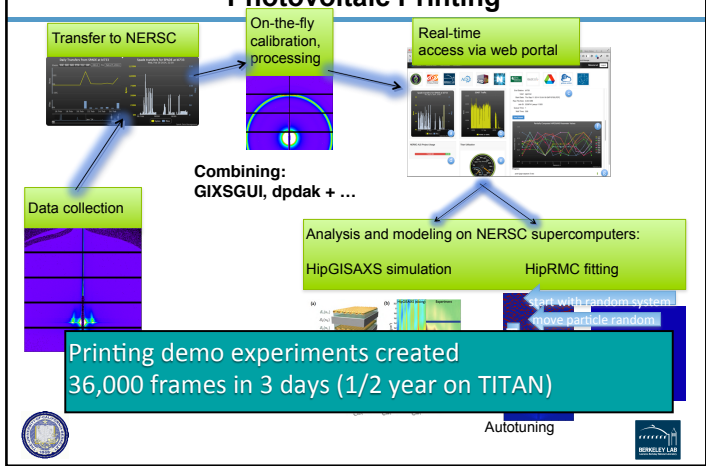
Make science easier, more reproducible, and democratic

Create a **"superfacility"** that integrates DOE Experimental facilities with computing centers and networking

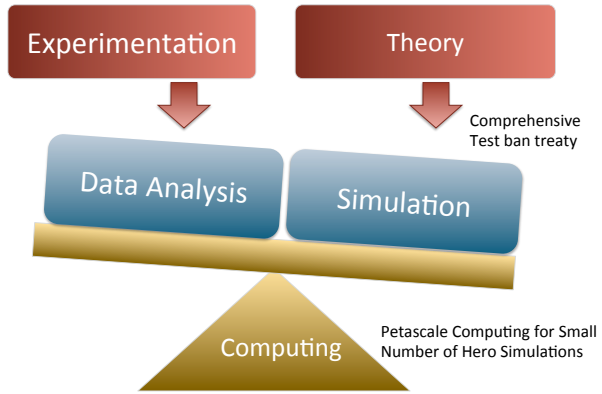
ESnet
ENERGY SCIENCES NETWORK

Data Demos 2014

A SuperFacility Demo for Light Sources / Photovoltaic Printing

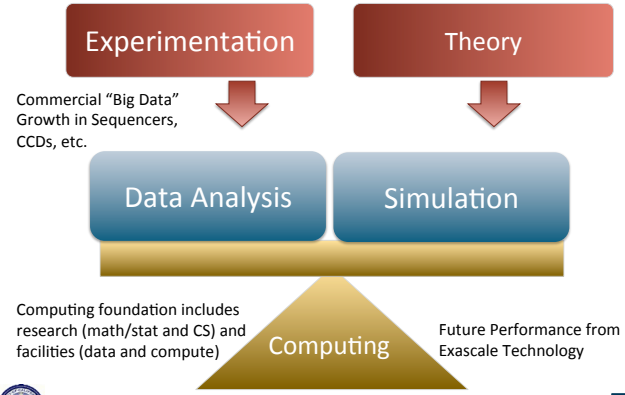


Advanced Computing: Not just for Simulation



- 22 -

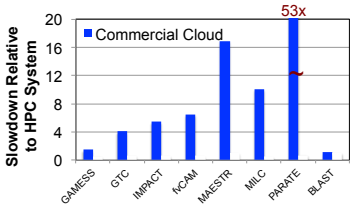
Science Needs Computing for Both Experiments (Data) and Theory (Modeling and Simulation)



23

Myth: Supercomputers are Expensive, Clouds are Cheap

Component	Annual Cost (rough estimate)
Cloud cost on apps (ave 5x slowdown)	~$900M
Cloud cost (1.38B core hours)	$181M
NERSC Budget	$57M
NERSC HPC HW	~$20M

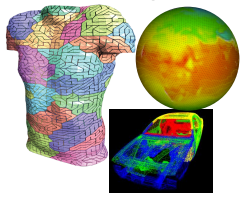


To buy raw NERSC core hours costs more than NERSC budget
- Even ignoring the measured performance slowdown
- **Doesn't include consulting staff, account management, licenses, bandwidth, software support: ~2/3 of NERSC's Budget**

Why?
- NERSC runs at higher utilization (> 90%) and **no profit.**
- NERSC cost/core hours dropped 10x (1000%) from 2007 to 2011, while Amazon pricing dropped 15% in the same period

Data Analytics: Case for PGAS

More Regular



Message Passing Programming

Divide up domain in pieces
Compute one piece
Send/Receive data from others

MPI, and many libraries

More Irregular

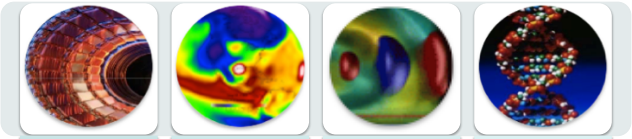
Global Address Space Programming

Each start computing
Grab whatever / whenever

UPC, CAF, X10, Chapel, GlobalArrays

25

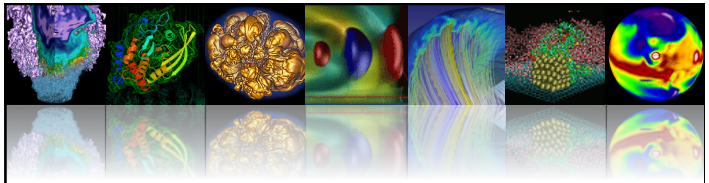
Programming Challenge? Science Problems Fit Across the "Irregularity" Spectrum



Massive Independent Jobs for Analysis and Simulations	Nearest Neighbor Simulations	All-to-All Simulations	Random access, large data Analysis

… often they fit in multiple categories

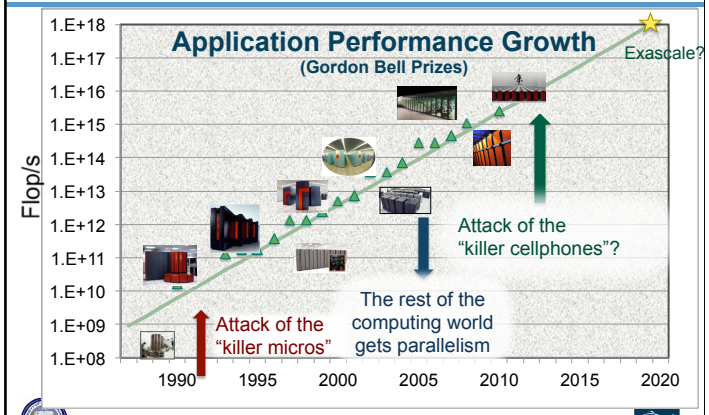
26



What about Exascale?

27

Computational Science has Moved through Difficult Technology Transitions

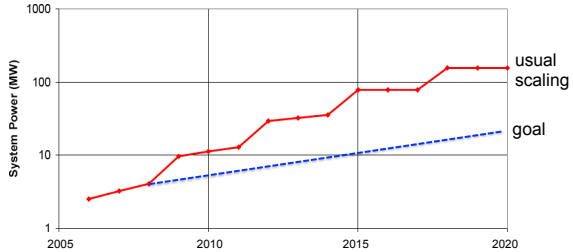


28

Energy Efficient Computing is Key to Performance Growth

At $1M per MW, energy costs are substantial

- **1 petaflop in 2010 used 3 MW**
- **1 exaflop in 2018 would use 100+ MW with "Moore's Law" scaling**



This problem doesn't change if we were to build 1000 1-Petaflop machines instead of 1 Exasflop machine. It affects every university department cluster and cloud data center.

29

"Exascale" Challenges Affect Performance Growth at all Scales

1) **Power** is the primary constraint
2) **Parallelism** (1000x today)
3) **Processor architecture** will change
4) **Data movement** dominates
5) **Memory** growth will not keep up
6) **Programming models** will change
7) **Algorithms** must adapt
8) **I/O** performance will not keep up
9) **Resilience** will be critical at this scale
10) **Interconnect bisection** must scale

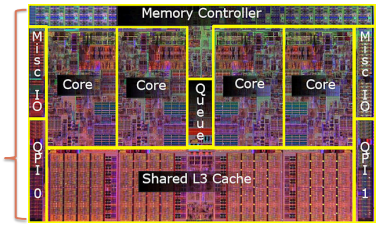
- **These are all at the node levels**
- **Happening NOW!**
- **Emerging Programming solutions are**
  - **Hard to use**
  - **Non-portable**
  - **Non-durable**

30

Challenge: New Processor Designs are Needed to Save Energy

Cell phone processor
(0.1 Watt, 4 Gflop/s)

Server processor
(100 Watts, 50 Gflop/s)



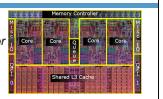
- **Server processors have been designed for performance, not energy**
  - **Graphics processors are 10-100x more efficient**
  - **Embedded processors are 100-1000x**
  - **Need manycore chips with thousands of cores**

31

Node Programming, Heterogeneity

- **Case for heterogeneity**
  - **Many small cores and SIMD for energy efficiency; few CPUs for OS / speed**
  - **Dark silicon → too many transistors to power**
- **Local store, explicitly managed memory**
  - **More efficient (get only what you need) and simpler hardware**
- **Split memory between CPU and "Accelerators"**
  - **Driven by market history and simplicity, but may not last**
  - **Communication: The bus is a significant bottleneck.**
- **Co-Processor interface between CPU and Accelerator**
  - **Default is on CPU, only run "parallel" code in limited regions**
  - **Why are the minority CPUs in charge?**

Cell phone processor (0.1 Watt, 4 Gflop/s)



Server processor (100 Watts, 50 Gflop/s)

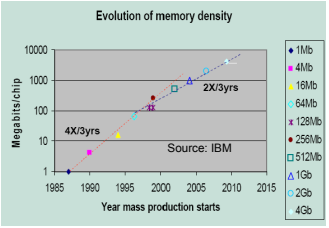
Is there a programming model that works for everyone?

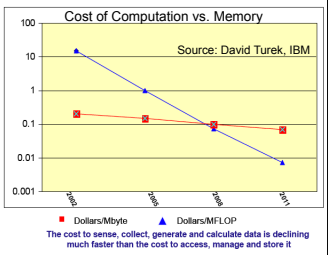
32

Challenge: Memory is Not Keeping Pace

Technology trends against a constant or increasing memory per core

- Memory density is doubling every three years; processor logic is every two
- Storage costs (dollars/Mbyte) are dropping gradually compared to logic costs



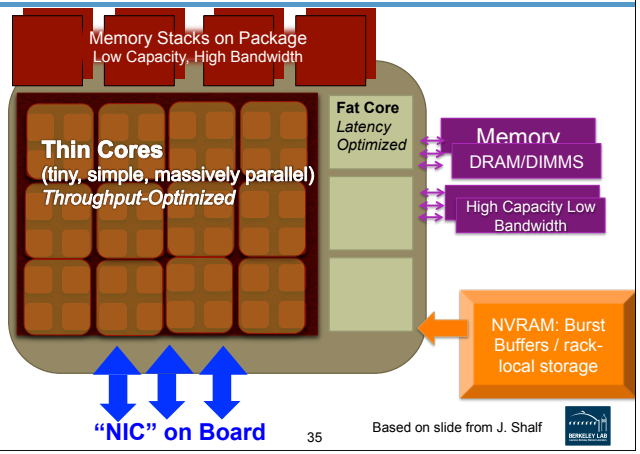


Question: *Can you double concurrency without doubling memory?*

The Memory ~~Wall~~ Swamp

Multicore didn't cause this, but kept the bandwidth gap growing.

Emerging Exascale Node Architecture



Based on slide from J. Shalf

Node Architecture Problems

- **Problems**
  - **Many slow cores with less memory per core**
  - **Wide SIMD (wide enough you can't ignore it)**
  - **Locality issues (NUMA)**
- **Possible problems**
  - ~~**Limited cache coherence?**~~
  - **Fat cores (heterogeneity)?**
  - **Fat cores in charge (co-processor / accelerator)**
  - **Scratchpad (local store) memory?**

Non-problem

- No caching == no problem (trivially coherent)
- PGAS hardware lesson: don't cache remote values
- MPI or/accelerator PGAS between domains will be fine

Node Programming for Homogenous Cores

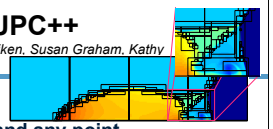
Requirements:
- Hardware exposes fast local accesses (minimize coherence)
- Low level software to control data layout and work assignment (pin to core)
- Algorithms that minimize data movement and overlap

Approach	Argument against
Flat MPI	Need different within/between node **algorithms**
MPI + MPI	Not enough memory per core
MPI + OpenMP	NUMA effects too strong, compilation too hard
MPI + PGAS (SPMD, C++)	Not tuned and not yet standard
MPI + TiDA (SPMD, F)	Not yet standard or tuned
MPI + Dynamic Tasking	Runtime overheads and poor locality control

37

Titanium Arrays in UPC++

Amir Kamil (previously Phil Colella, Paul Hilfinger, Alex Aiken, Susan Graham, Kathy Yelick and many others)

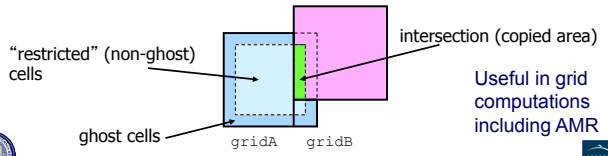


- **Key features of Titanium arrays**
  - **Generality: indices may start/end and any point**
  - **Domain calculus allow for slicing, subarray, transpose and other operations without data copies**
- **Use domain calculus to identify ghosts and iterate:**

```
foreach (p in gridA.shrink(1).domain()) ...
```

- **Array copies automatically work on intersection**

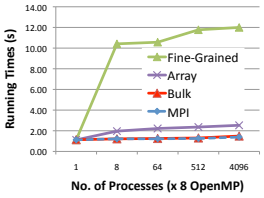
```
gridB.copy(gridA.shrink(1));
```

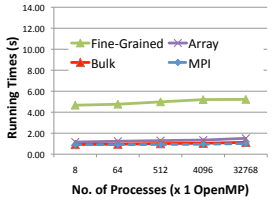


Useful in grid computations including AMR

Titanium Arrays in UPC++

Amir Kamil (previously Phil Colella, Paul Hilfinger, Alex Aiken, Susan Graham, Kathy Yelick and many others)





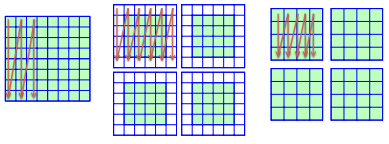

```
ndarray<double, 3, global> bArray =
  bArrays[PT(level, id, dir, i, j, k)];
bArray.async_copy(aArrays[PT(level, id, dir, i, j, k)]);
```

- **In the UPC++ array library, the same concepts are supported**
- **The syntax (without compiler support) is not as elegant**
- **The performance is close to that of a version that explicitly packs/unpacks (bulk) and to MPI**
- **The flat (no OpenMP) version is faster than hybrid**

39

TiDA: Tiling as a Durable Abstraction

Didem Unat, Cy Chan, Weiqun Zhang, John Bell, John Shalf




```
1  type(tile) :: tl
2  integer    :: tileno, tlo(2), thi(2), i, j
3  double precision, pointer :: ptrA(:,:)
4
5  do tileno=1, ntiles(tilearr)
6
7     ptrA => dataptr(A, tileno)
8     tl = get_tile(tilearr, tileno)
9     tlo = get_lwb(tl)
10    thi = get_upb(tl)
11
12    do j=tlo(2),thi(2)      !element loop 1
13       do i=tlo(1), thi(1) !element loop 2
14          !loop body
15          ptrA(i,j) = do_something(i,j)
16       end do
17    end do
18
19 end do !end of tile loop
```

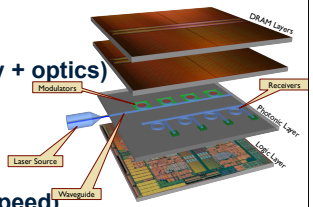
- **Tiling: Add loop nests so inner ones fit in cache, e.g., 3-loop matmul → 6-loop**
- **TiDA: Add tile shape/size information to each array**
- **Optionally change the data layout to match**
- **Can also add ghost regions as needed**

40

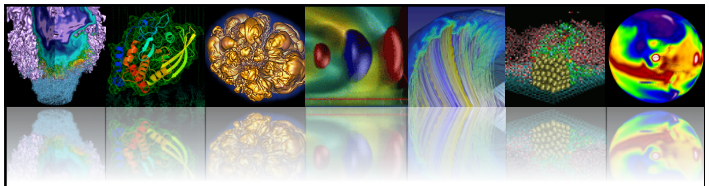
Memory Technology (Sandia, Micron, Columbia LBNL)

Understand the Potential of Intelligent, Stacked DRAM Technology

- Data movement are projected to account for over 75% of power on an exascale platform
- Work to reduce that via
  - Optical interconnect(s)
  - 3D stacking (logic + memory + optics)
  - New memory protocols



- Research Questions
  - What is the performance of stacked memory (power & speed)
  - How much intelligence to put into logic layer
    - Atomics, gather/scatter, checksums, full-processor-in-memory
  - What is the memory consistency model
  - How to program it ?

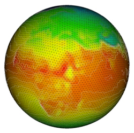


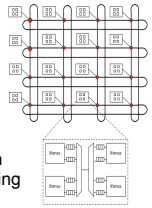
Co-Design architectures for Science

Keeping in mind market pressures

Co-Design in the Green Flash Project

- Demonstrated during SC '09
- CSU atmospheric model ported to low-power core design
  - Dual Core Tensilica processors running atmospheric model at 25MHz
  - MPI Routines ported to custom Tensilica Interconnect
- Memory and processor Stats available for performance analysis
- Emulation performance advantage
  - 250x Speedup over merely function software simulator
- *Actual code running - not representative benchmark*



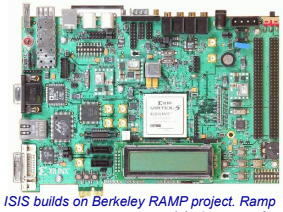


Icosahedral mesh for algorithm scaling

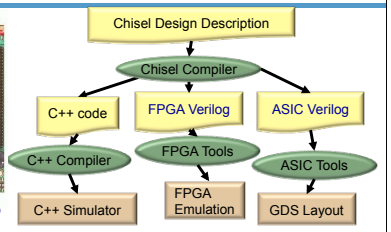
John Shalf, Dave Donofrio, Lenny Oliker, Michael Wehner, Marghoob Mohiyuddin, Shoaib Kamil

43

Enabling Manycore Architecture Research



ISIS builds on Berkeley RAMP project. Ramp Gold shown here which models 64 cores of SPARC v8 with shared memory on $750 board. Has hardware FPU, MMU; boots OS.



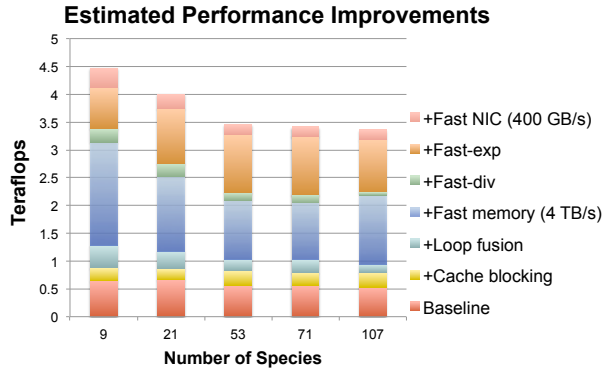
ISIS Hardware description language based on Scala, modern OO/Functional language that compiles to JVM.

- ISIS: rapid, accurate FPGA emulation of manycore chips
- Spans VLSI design and simulation and includes chip fab
  - Trains students in real design trade-offs, power and area costs
- Mapping RTL to FPGAs for algorithm/software co-design
  - 100x faster than software simulators and more accurate

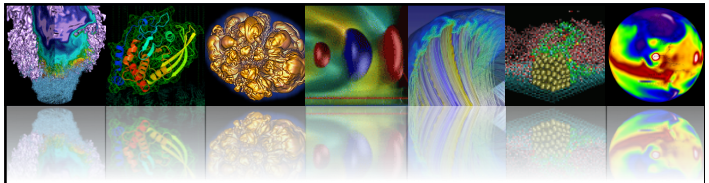
PIs: John Wawrzynek and Krste Asanovic, UC Berkeley

44

Co-Design Analysis for Combustion



- Hardware and software need to change together

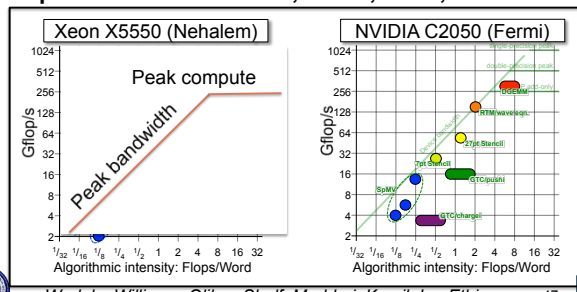


Let computers, not humans, tune for modern architectures code

But beware of trusting compilers

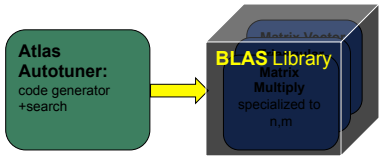
Autotuning: Write Code Generators

- Autotuners are code generators plus search
- Avoids two unsolved compiler problems: dependence analysis and accurate performance models
- Popular in libraries: Atlas, FFTW, OSKI,…



Work by Williams, Oliker, Shalf, Madduri, Kamil, Im, Ethier,… 47

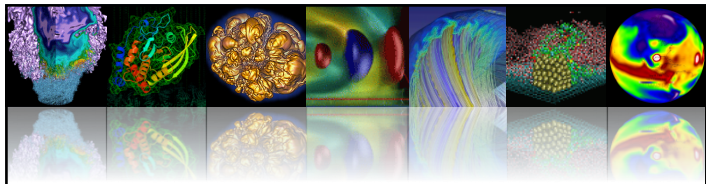
Approaches to Autotuning



How do we produce all of these (correct) versions?

- Using scripts (Python, perl, C,..)
- Transform high level representation (FFTW, Spiral)
- Compiling a domain-specific language (D-TEC)
- Compiling a general-purpose language (X-Tune)
- Dynamic compilation of a domain-specific (SEJITS)

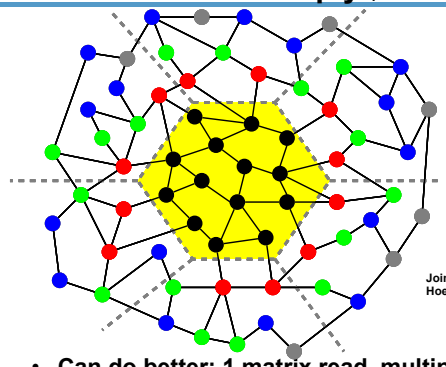
48



Target Higher Level Loops

Harder than inner loops....

Iterative Solves are Dominated by Sparse Matrix-Vector Multiply (nearest neighbor on graph)



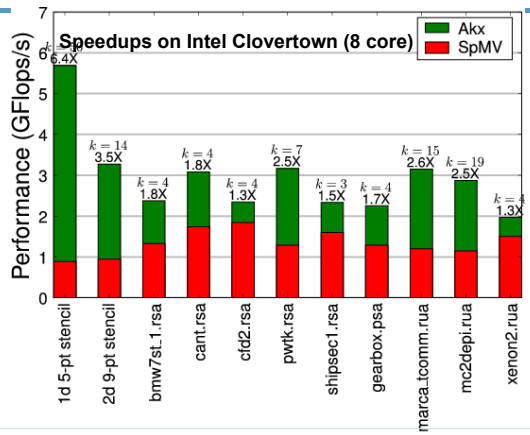
For implicit memory management (caches) uses a TSP algorithm for layout

Joint work with Jim Demmel, Mark Hoemman, Marghoob Mohiyuddin

- **Can do better: 1 matrix read, multiple multiplies**
  - **Serial: O(1) moves of data moves vs. O(k)**
  - **Parallel: O(log p) messages vs. O(k log p)**

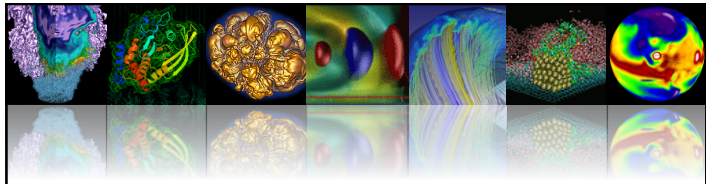
50

Bigger Kernel ($A^k x$) Runs at Faster Speed than Simpler (Ax)



Jim Demmel, Mark Hoemmen, Marghoob Mohiyuddin, Kathy Yelick

51

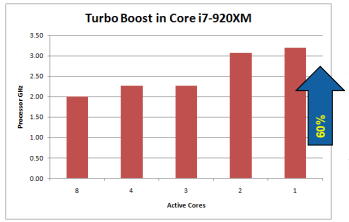


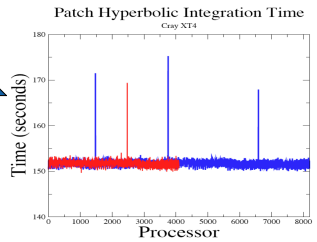
Avoid Synchronization

The end of bulk-synchronous programming?

Reasons to avoid synchronization

- **Processors do not run at the same speed**
  - Never did, due to caches
  - Power / temperature management makes this worse

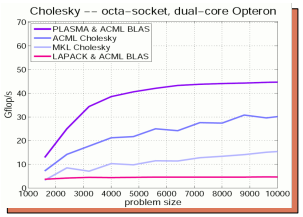




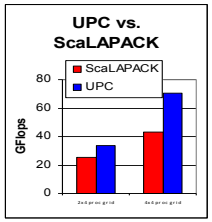
53

DAG Scheduling Outperforms Bulk-Synchronous Style

PLASMA on shared memory



UPC on partitioned memory



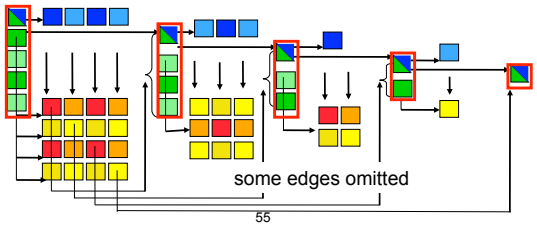
UPC LU factorization code adds cooperative (non-preemptive) threads for latency hiding
- New problem in partitioned memory: allocator deadlock
- Can run on of memory locally due tounlucky execution order

PLASMA by Dongarra et al; UPC LU joint with Parray Husbands

54

Event Driven LU in UPC

- Assignment of work is static; schedule is dynamic
- Ordering needs to be imposed on the schedule
  - Critical path operation: Panel Factorization
- General issue: dynamic scheduling in partitioned memory
  - Can deadlock in memory allocation
  - "memory constrained" lookahead



55

One-sided communication is a mechanism that works everywhere

PGAS is a programming model

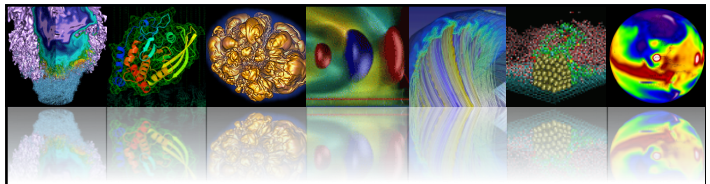
```
*p1 = *p2 + 1;
A[i] = B[i];

upc_memput(A,B,64);
```

Uses 1-sided communication: put/get

This Direct Memory Access (DMA) also appears in:
- Fast one-sided network communication (RDMA, Remote DMA)
- Move data to/from accelerators
- Move data to/from I/O system (Flash, disks,..)
- Movement of data in/out of local-store (scratchpad) memory

56

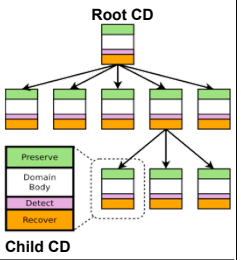


Resilience

Is the sky really falling?

Resilience Approaches

- **Containment Domains (CDs) for trees**
  - Flexible resilience techniques (mechanism not policy)
  - Each CD provides own recovery mechanism
  - Analytical model: 90%+ efficiency at 2 EF vs. 0% for conventional checkpointing
- **Berkeley Lab Checkpoint Restart**
  - BLCR is a system-level Checkpoint/Restart
    - Job state written to filesystem or memory; works on most HPC apps
  - Checkpoint/Restart can be used for roll-back recovery
    - a course-grained approach to resilience
    - BLCR also enables use for job migration among compute nodes
  - Requires support from the MPI implementation
- **Impact: part of standard Linux**



- ***Preserve*** data on domain start
- ***Compute*** *(domain body)*
- ***Detect*** faults before commit
- ***Recover*** from detected errors

CD PIs: Mattan Erez (+Eric Roman for PGAS); GVR PI: Andrew Chien

59

What is Wrong with Current Operating Systems?

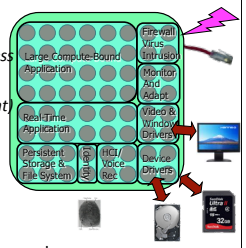
Tesselation: Joint UCB/LBNL to rethink Manycore OSs

Assumes limited number of CPUs that must be shared
- ***Old CW***: *time-multiplexing*
- ***Tesselation:*** *spatial partitioning*

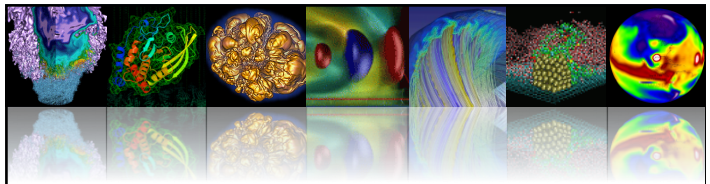
Greedy allocation of finite I/O device interfaces
- ***Old CW:*** *First process to acquire lock gets device*
- ***Tesselation:*** *QoS management for symmetric device access*

Fault Isolation
- ***Old CW:*** *CPU failure → Kernel Panic (increasingly frequent)*
- ***Tesselation:*** *CPU failure → Partition Restart (w/ drivers)*

Inter-Processor Communication
- ***Old CW:*** *invoked for ANY interprocessor communication*
- ***Tesselation:*** *direct HW access mediated by hypervisor*

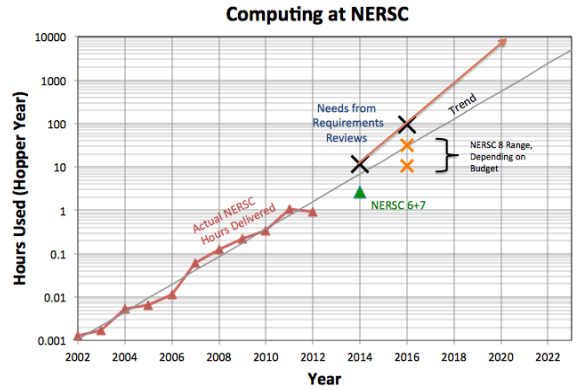


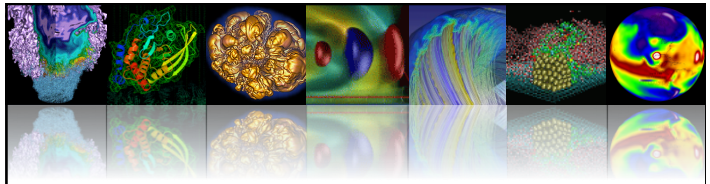
Impact:
- Convex optimization major thrust for Microsoft Research
- Launching into new OS/R CFP with Sandia National Lab



What does this mean for NERSC?

Keeping up with user needs will be a challenge





Essentially, all models are wrong, but some are useful.

-- George E. Box, Statistician

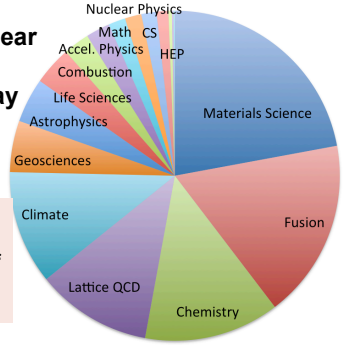
NERSC Workload

- **> 5000 users**
- **> 700 projects**
- **> 2000 publications per year**

- **2 Petascale systems today**
  - **NERSC-7: Hopper**
  - **NERSC-8: Edison**
- **Moving data workload**

The workload is diverse and increasingly complex due to science workflows, integration of data, and demand for higher resolution and scale

2013 Breakdown of Allocations by Science Area



Edison, a Cray XC-30 plays a key role in NERSC's strategy

- NERSC assessed that our broad workload was not ready for GPUs and procured Edison, with Ivy Bridge Intel CPUs
- Workloads that have difficulty moving to NERSC-8 can still work productively on Edison while the code is adapted
- In 2016 Edison will likely provide ~20% of NERSC's cycles

- 65 -

The Cori System Scheduled for 2016

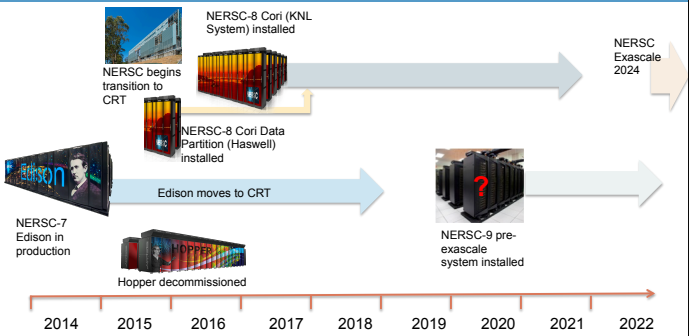
- **Cori will support the broad Office of Science research community transition to energy efficient architectures**
- **Cray XC system with > 9300 Intel KNL nodes**
  - Self-hosted (not an accelerator) manycore processor with over 60 cores per node
  - 32 Flops / cycle (AVX 512 SIMD)
  - On-package high-bandwidth memory (scratchpad)
  - Scheduled for 2016 installation
- **"Data Partition" with ~2000 Haswell nodes (2015)**
  - NVRAM Burst Buffer for data intensive applications
  - 28 PB of disk, 432 GB/sec I/O bandwidth
  - Scheduling for complex workflows

Image source: Wikipedia

System named after Gerty Cori, Biochemist and first American woman to receive the Nobel prize in science.

- 66 -

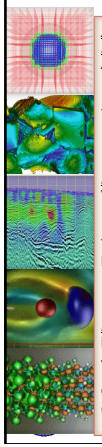
NERSC System Roadmap



- **NERSC Exascale Strategy is designed to balance the needs of current science with future science**

- 67 -

NESAP Codes



Advanced Scientific Computing Research

Almgren (LBNL) **BoxLib AMR**

Framework

Trebotich (LBNL) **Chombo-crunch**

High Energy Physics

Vay (LBNL) **WARP & IMPACT**

Toussaint(Arizona) **MILC**

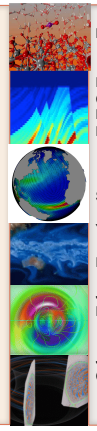
Habib (ANL) **HACC**

Nuclear Physics

Maris (Iowa St.) **MFDn**

Joo (JLAB) **Chroma**

Christ/Karsch (Columbia/BNL) **DWF/HISQ**



Basic Energy Sciences

Kent (ORNL) **Quantum Espresso**

Deslippe (NERSC) **BerkeleyGW**

Chelikowsky (UT) **PARSEC**

Bylaska (PNNL) **NWChem**

Newman (LBNL) **EMGeo**

Biological and Environmental Research

Smith (ORNL) **Gromacs**

Yelick (LBNL) **Meraculous**

Ringler (LANL) **MPAS-O**

Johansen (LBNL) **ACME**

Dennis (NCAR) **CESM**

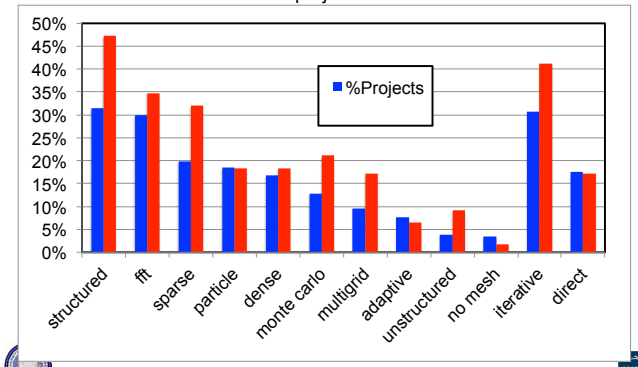
Fusion Energy Sciences

Jardin (PPPL) **M3D**

Chang (PPPL) **XGC1**

Numerical Methods at NERSC

- Quantitative (but not so deep) measure of algorithms classes
- Based on hours allocated to a project that the PI claims uses the method



69

Algorithm Diversity

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

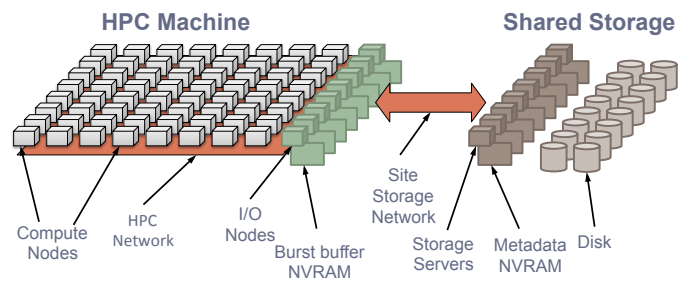
NERSC Qualitative In-Depth Analysis of Methods by Science Area

Previous Procurement Strategy: Publish Representative Benchmarks

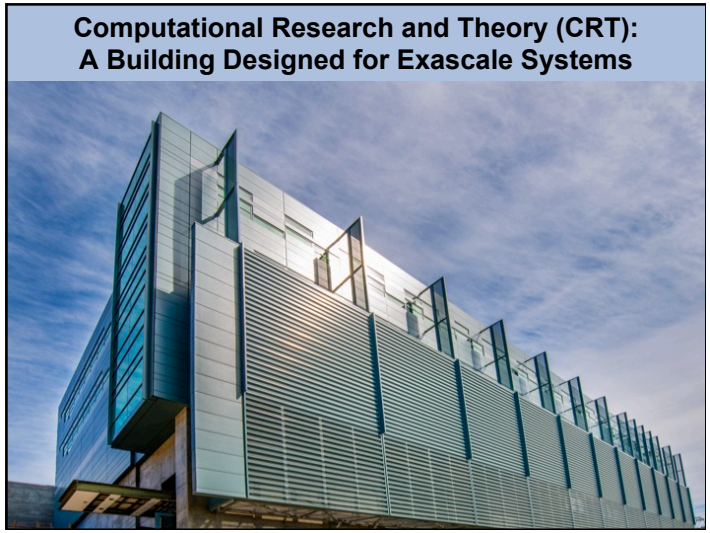
Science areas	Dense linear algebra	Sparse linear algebra	Spectral Methods (FFT)s	Particle Methods	Structured Grids	Unstructured or AMR Grids
Accelerator Science		X	X IMPACT-T	X IMPACT-T	X IMPACT-T	X
Astrophysics	X	X MAESTRO	X	X	X MAESTRO	X MAESTRO
Chemistry	X GAMESS	X	X	X		
Climate			X CAM		X CAM	X
Fusion	X	X		X GTC	X GTC	X
Lattice Gauge		X MILC	X MILC	X MILC	X MILC	
Material Science	X PARATEC		X PARATEC	X	X PARATEC	

71

Co-design for Data: Finding Middle Ground



- **Mount BB as a disk: /fast – then user has to do all the work/juggling**
- **Have software that automatically determines best way to use BB - $'s**


Computational Research and Theory (CRT):
A Building Designed for Exascale Systems