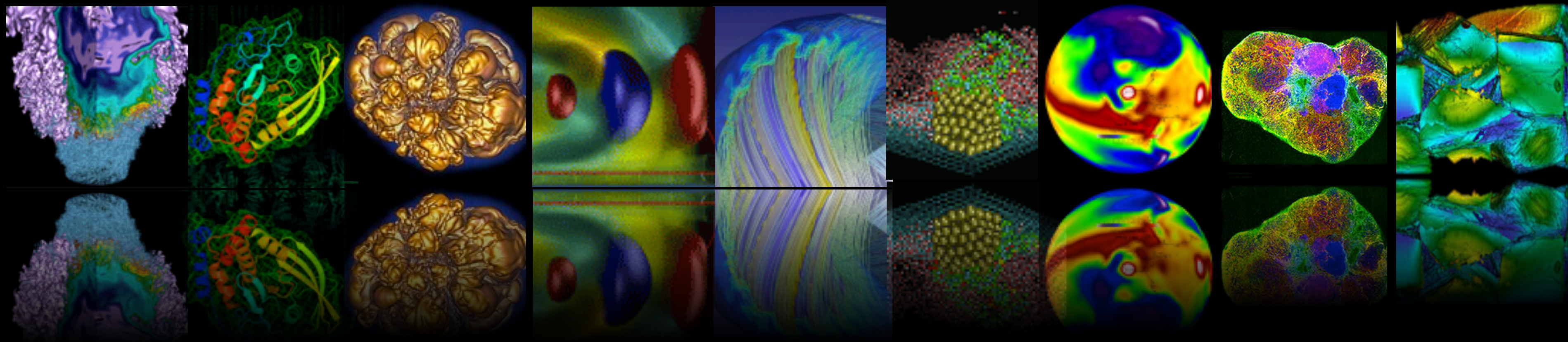
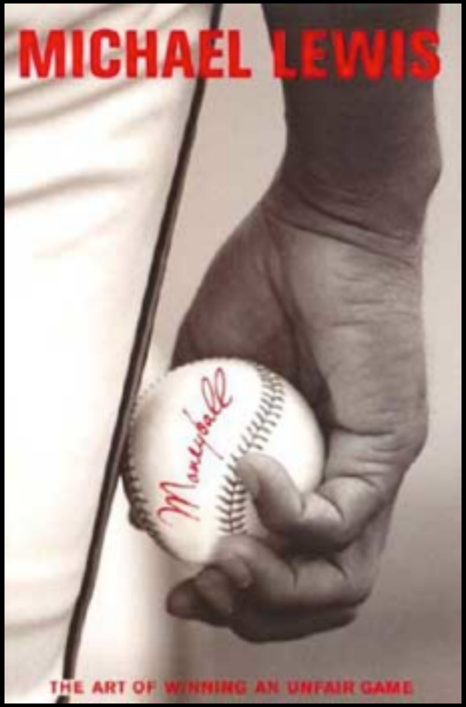


How to Teach your Exascale Machine to Do the Data Dance

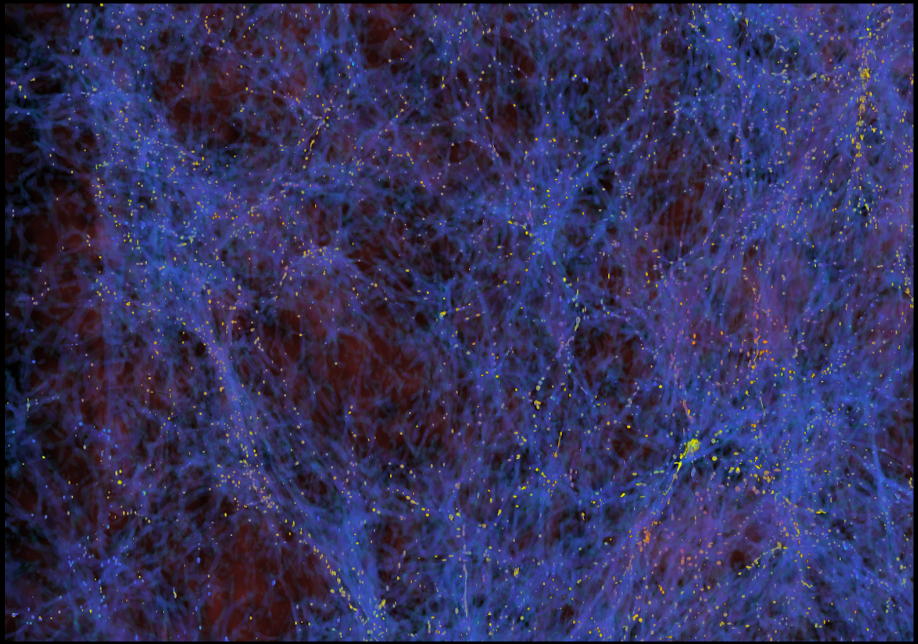
Kathy Yelick

Professor of Electrical Engineering and Computer Sciences
University of California at Berkeley
Associate Laboratory Director for Computing Sciences
Lawrence Berkeley National Laboratory

“Big Data” Changes Everything…What about Science?



Combine simulation and observation for next Cosmology breakthrough

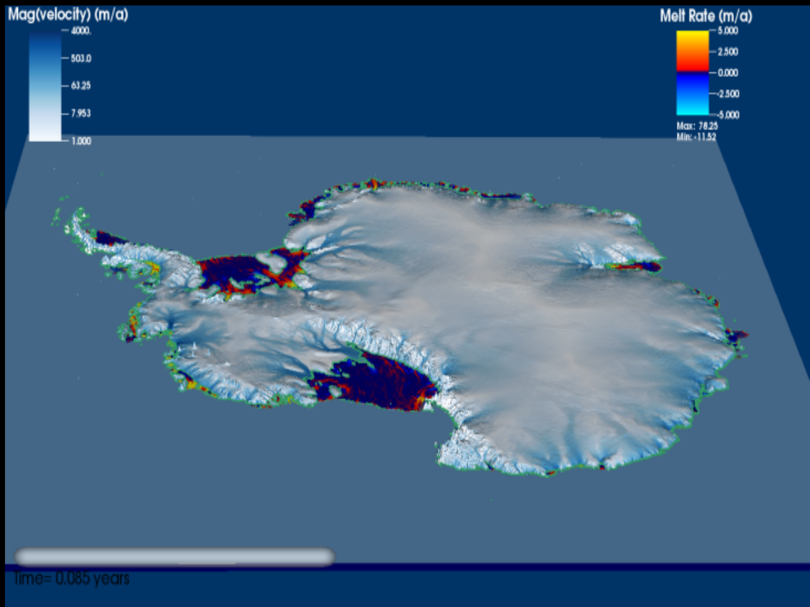


Nyx simulation of Lyman alpha forest using AMR

Kitt Peak National Observatory's Mayall 4-meter telescope, planned site of the DESI experiment

Reduce systematic bias in observation through simulation of ~1 Gigaparsec Baryon Acoustic Oscillations in the Lyman Alpha Forest and ~100 Gigaparsec simulation of galaxy clusters, both requiring adaptive mesh refinement (AMR).

Climate models and microbial analysis together to predict the future of the environment



New climate modeling methods, including AMR "Dycore" produce new understanding of ice

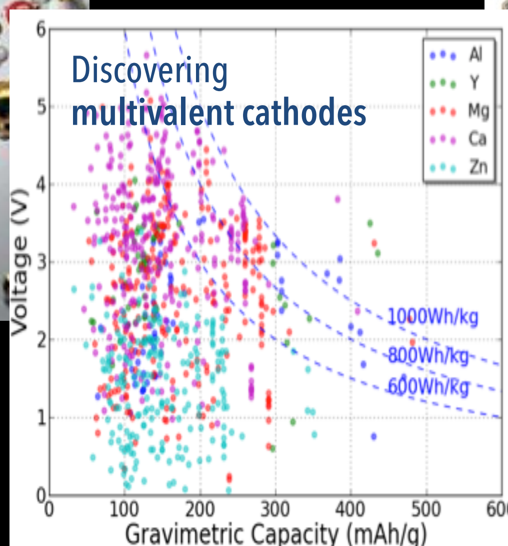
Genomes to watersheds Scientific Focus Area

Understand interactions between environmental microbiomes and climate change with *kilometer resolution models* that track dynamic 3D features (with AMR) and *genome-enabled analysis* of environmental sensors.

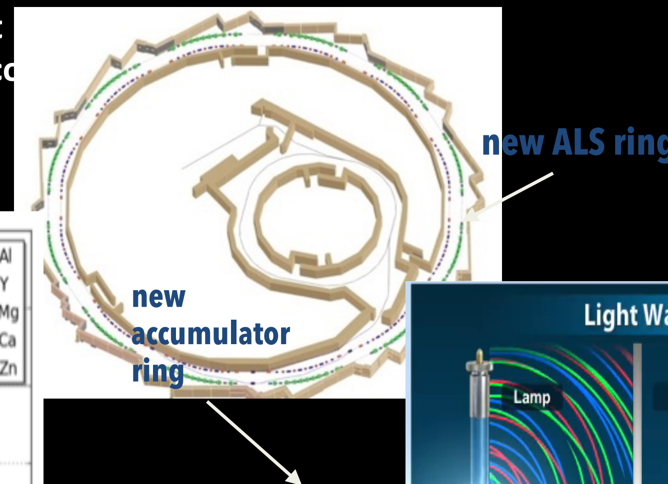
Understand and control energy with advanced light sources and materials modeling

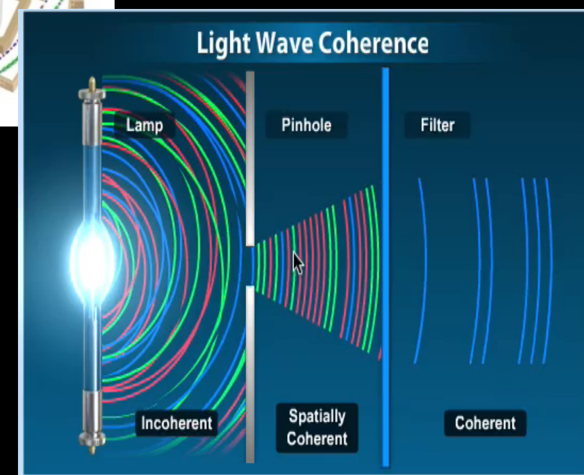
Materials Project

13,030 users hosted at NERSC with software co developed by CRD



ALS-U Upgrade

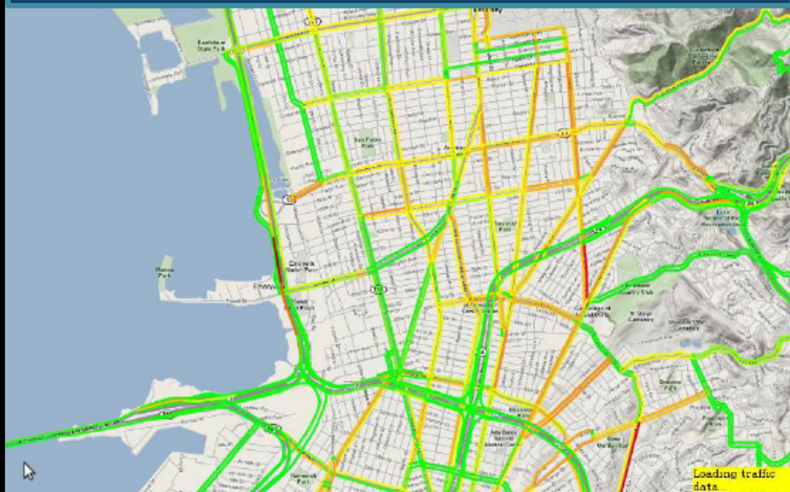




Understand and control the direction and flow of energy with minimal losses using *advanced instruments*, *high fidelity models*, and high throughput simulation and analysis for applications in energy, environment and computing,

Science in embedded sensors: Internet of Things

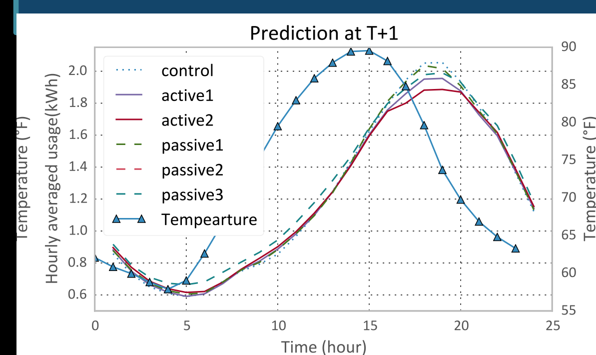
Transportation Modeling



Power Grid Modeling

Scenario Prediction, Planning

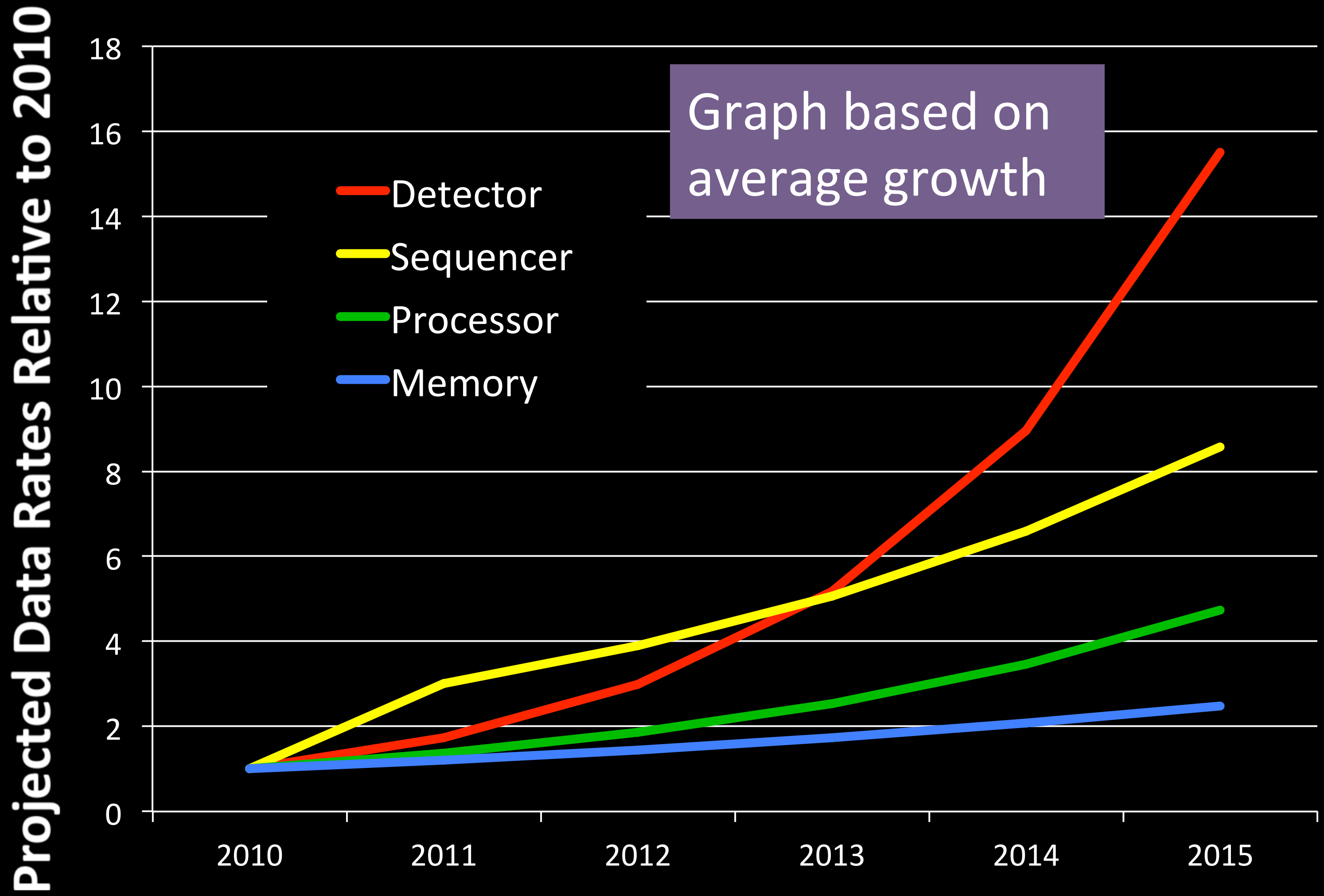
Decision Science



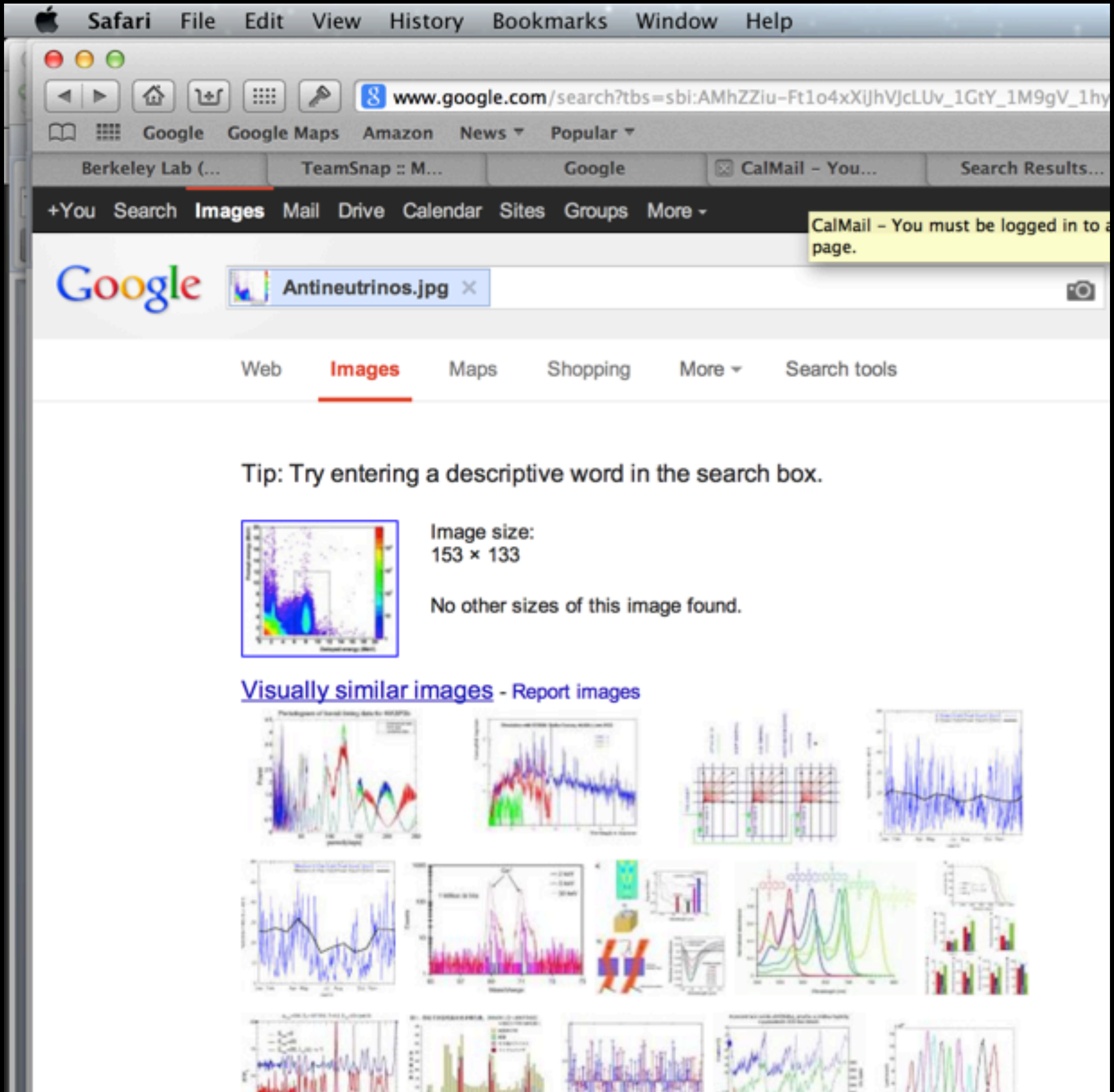
Roadmap for this talk

- ✓ Science at the boundary of simulation and observation
- Science data challenges
- What do I mean by Exascale?
- Exascale challenges for Data problems
  - Programming models
  - Algorithms
  - Architectures
  - Systems
  - Policies

Science Data Growth is Outpacing Computing

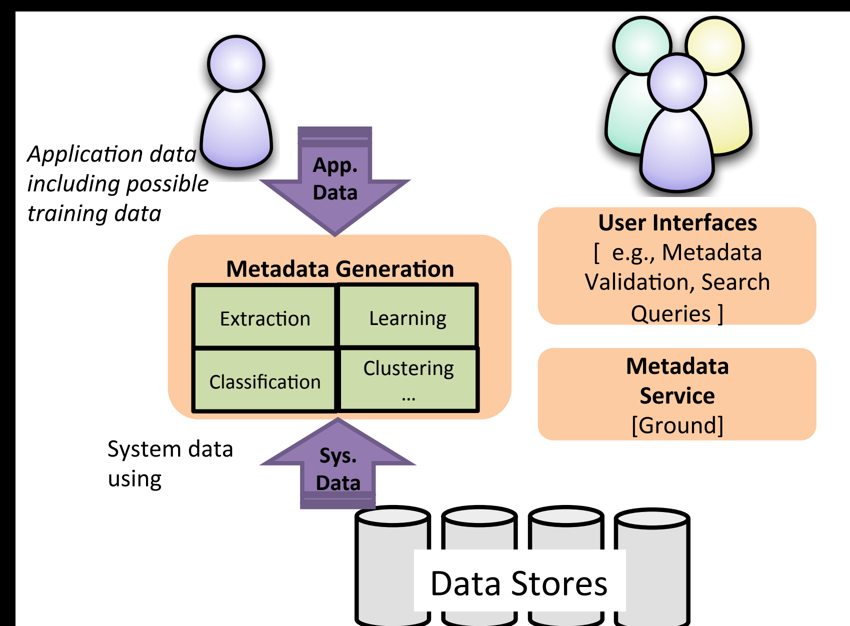


Old School Scientific Data Search



Automated Search, Meta-Data Analysis, and On-Demand Simulation

Automated metadata extraction using machine learning



Jobs submitted by “bots” based on queries; algorithms extract informatics for design

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

Roadmap for this talk

- ✓ Science at the boundary of simulation and observation
- ✓ Science data challenges
- What do I mean by Exascale?
- Exascale challenges for Data problems
  - Programming models
  - Algorithms
  - Architectures
  - Systems
  - Policies

Exascale = Orders of magnitude increase in performance at all scales

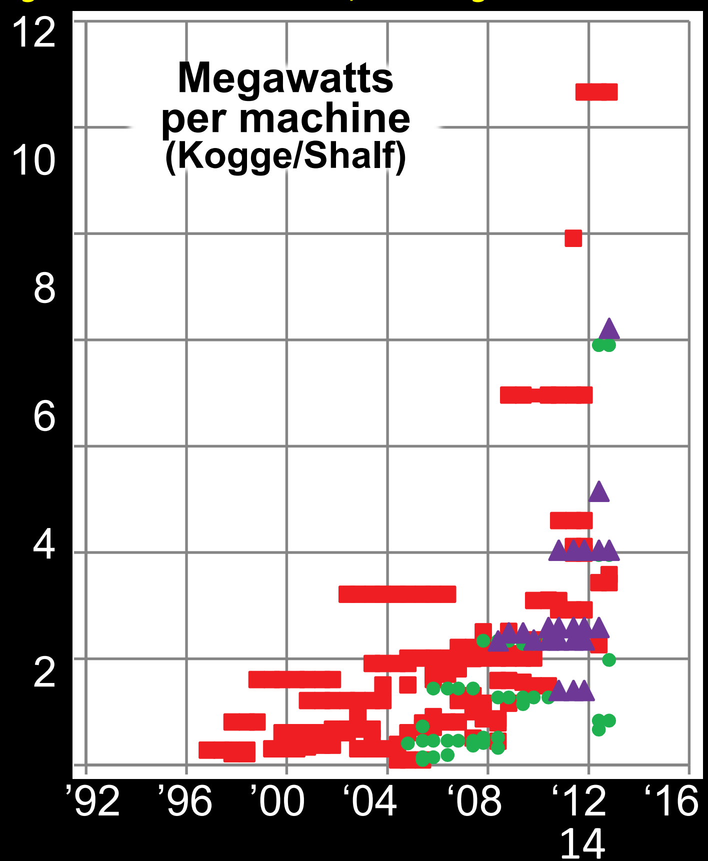
→ Continued growth at constant energy

→ Continued growth at constant cost

Computing is energy-constrained

At ~$1M per MW, energy costs are substantial
- 1 petaflop in 2008 used 3 MW
- 1 exaflop in 2018 at 200 MW “usual **chip** scaling”

Missing Tihanhe-2 at 18MW, Taihulight at 15MW

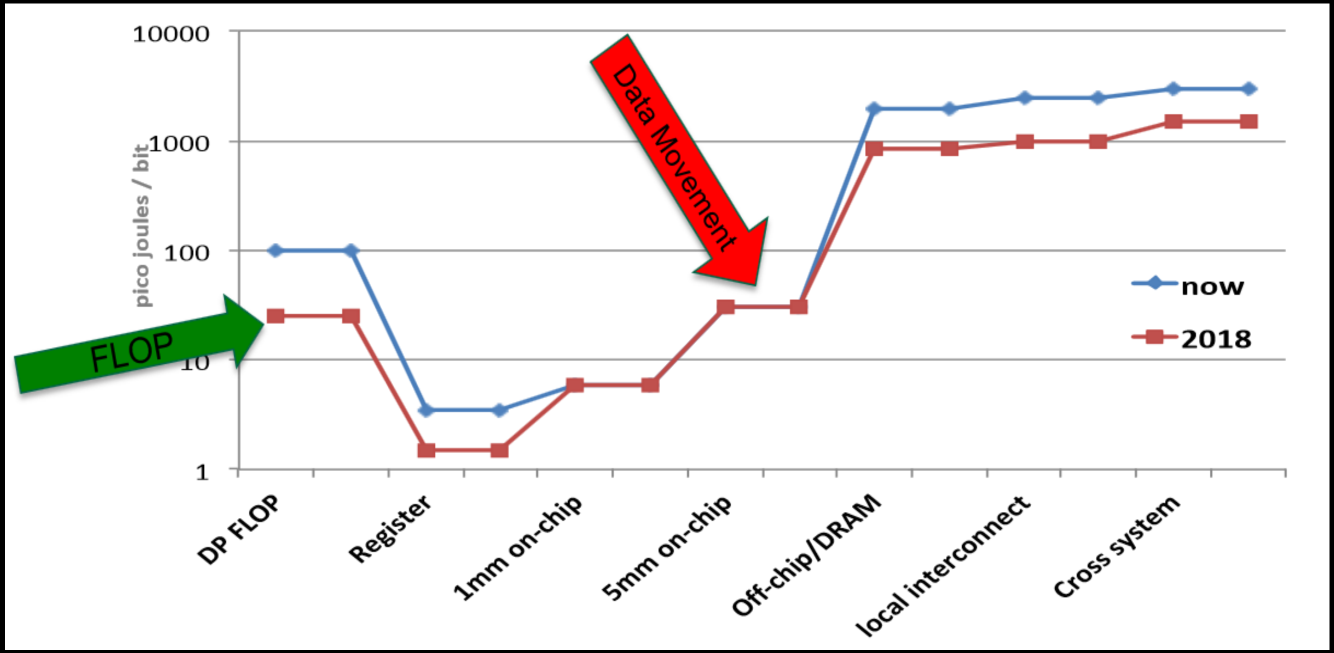


Goal: 1 Exaflop in 20 MW
= 20 pJ / operation

Note: The 20 pJ / operation is
- **Independent of machine size**
- **Independent of # cores used per application**
- **But “operations” need to be useful ones**

Challenge: Communication is expensive

Communication is expensive in time and energy



Hard to change: Latency is physics; bandwidth is money!

Roadmap for this talk

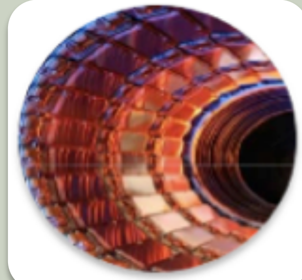
✓ **Science at the boundary of simulation and observation**

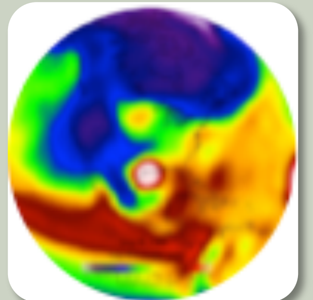
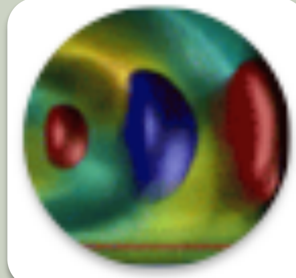
✓ **Science data challenges**

✓ **What do I mean by Exascale?**

- **Exascale challenges for Data problems**
  - Programming models
  - Algorithms
  - Architectures
  - Systems
  - Policies

Data vs. Simulation: The Irregularity Spectrum




Massive Indepen-dent Jobs	Compute-Intensive	Nearest Neighbor	All-to-All	Random access, large data

Different architectures? Programming models?

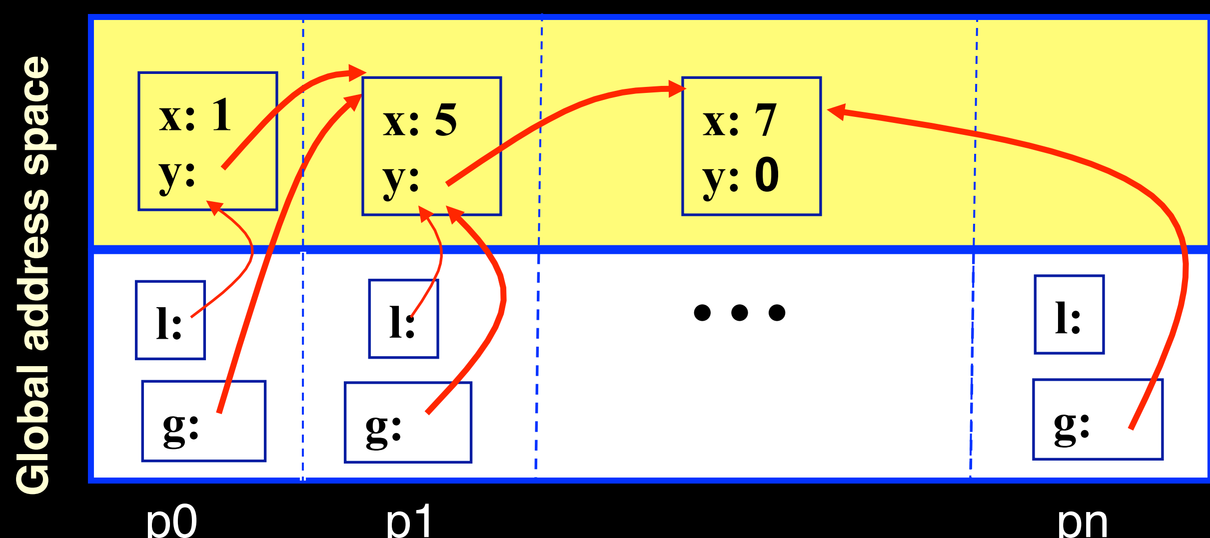
PGAS: A programming model for exascale

- ***Global address space:*** thread may directly read/write remote data using an address (pointers and arrays)

  … = *gp;   ga[i] = …

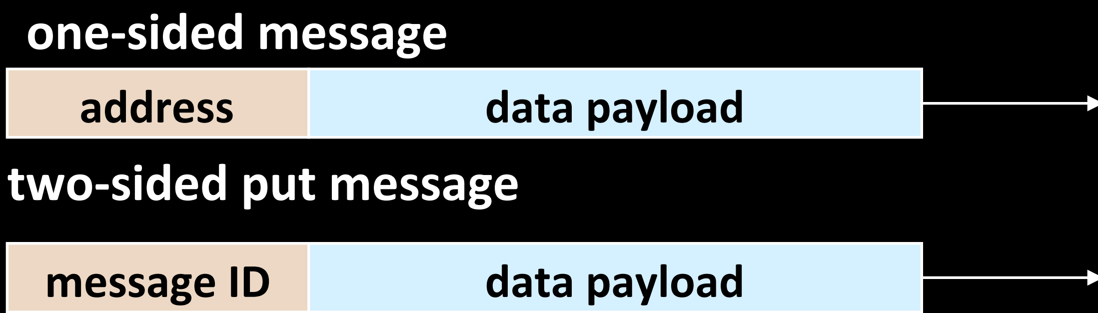
- ***Partitioned:*** data is designated as local or global

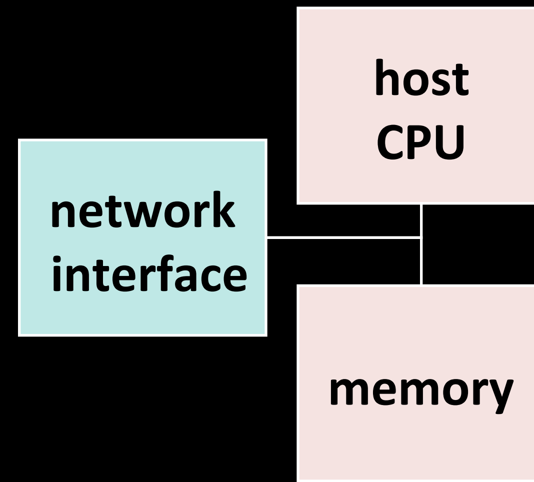
  shared int [ ] ga; and upc_malloc (…)



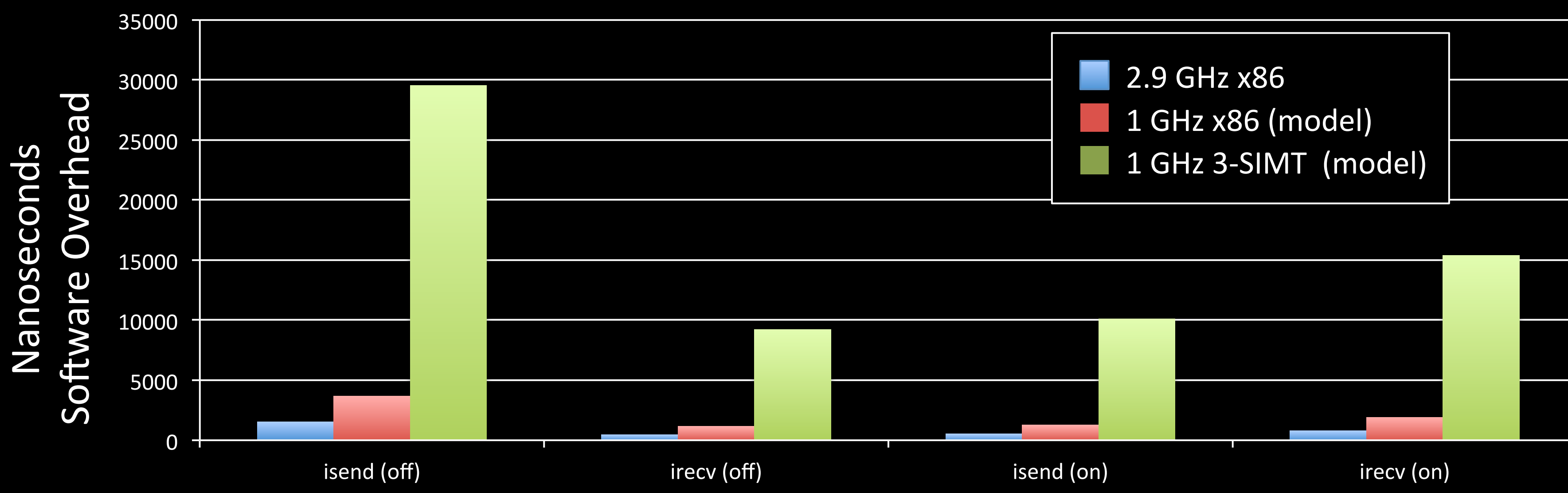
A programming model can influence how programmers think

One-Sided Communication is Closer to Hardware

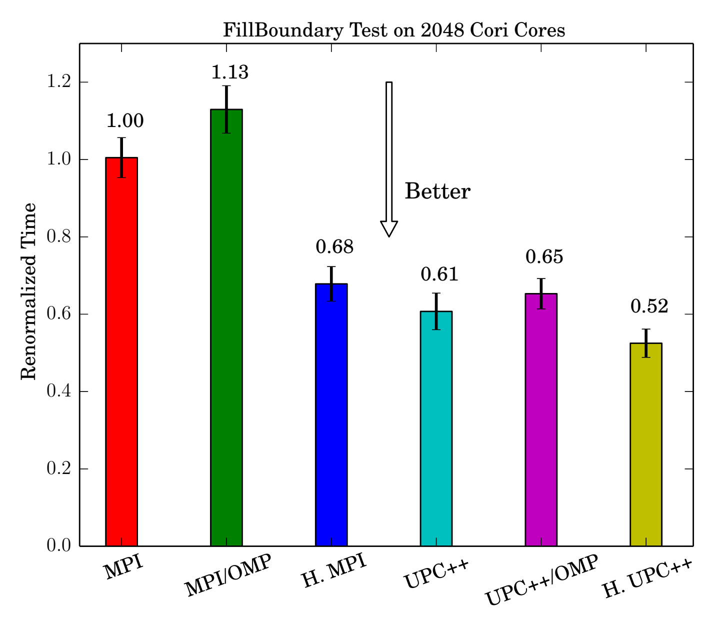


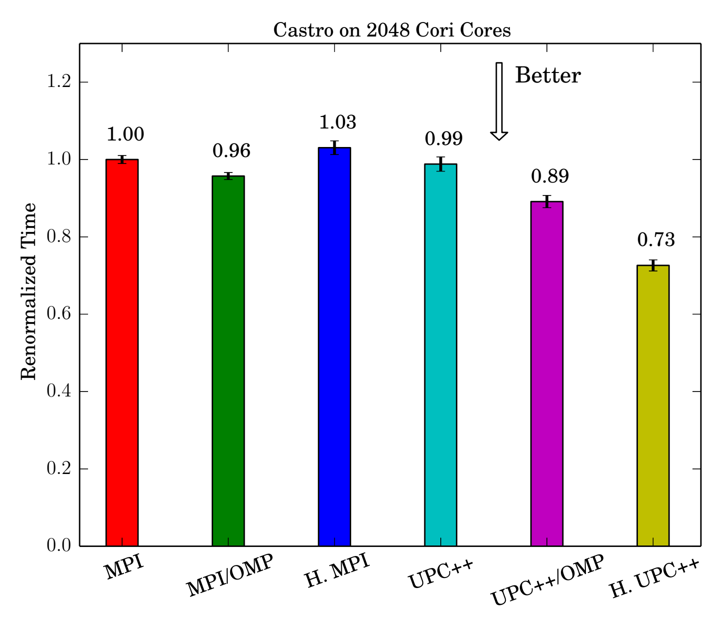


- Hardware does 1-sided communication
- Overhead for send/receive messaging is worse at exascale



One-sided PGAS (UPC++) in AMR





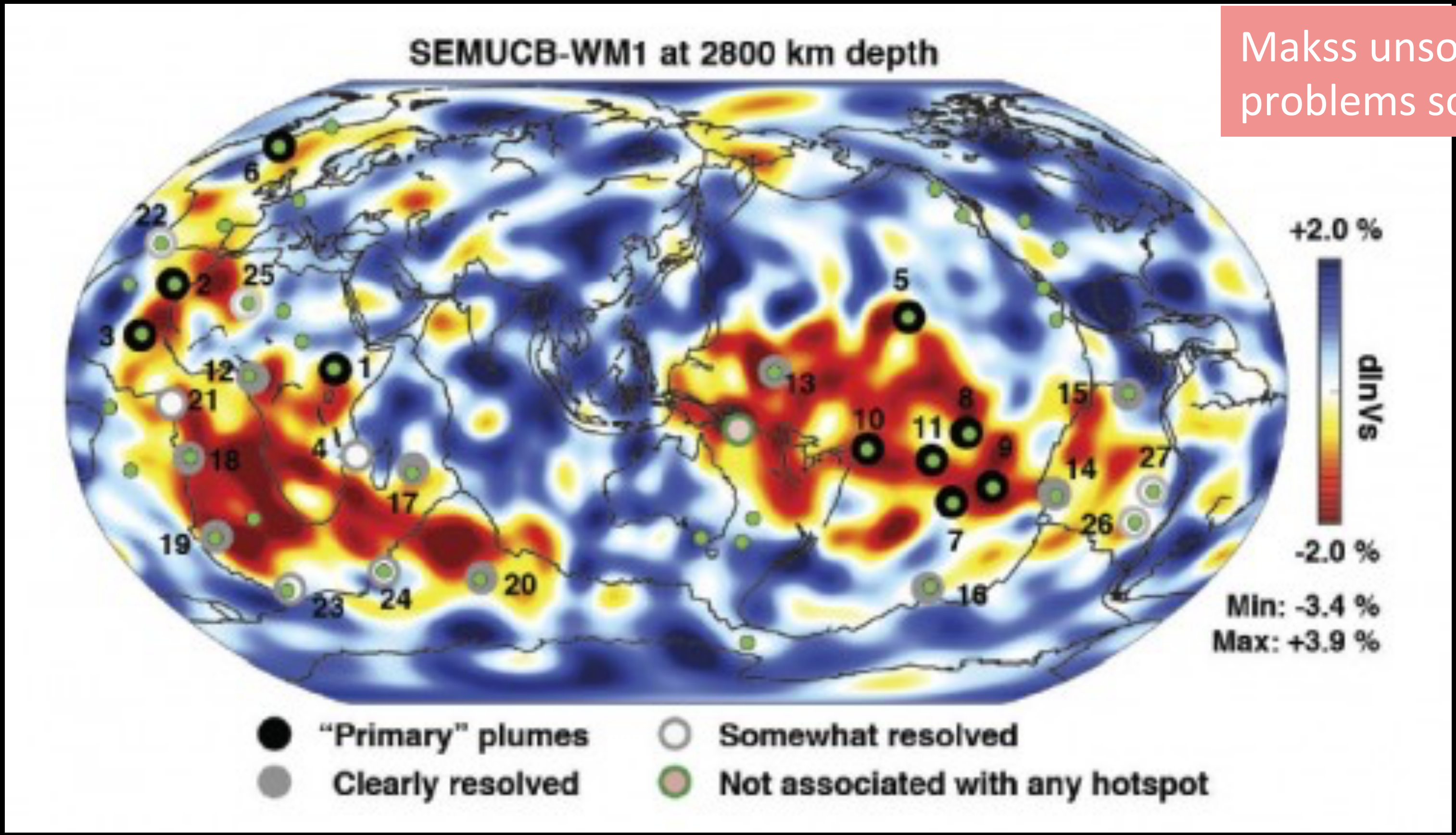
- **Adaptive Mesh Refinement (AMR) using UPC++**
  - Metadata costs make flat MPI impractical
  - Replaced communication (retained most code)
  - Hierarchical algorithms (UPC++/UPC++ or MPI/MPI best)

Weiquin Zhang, Y. Zheng

Science Impact: Whole-Mantle Seismic Model

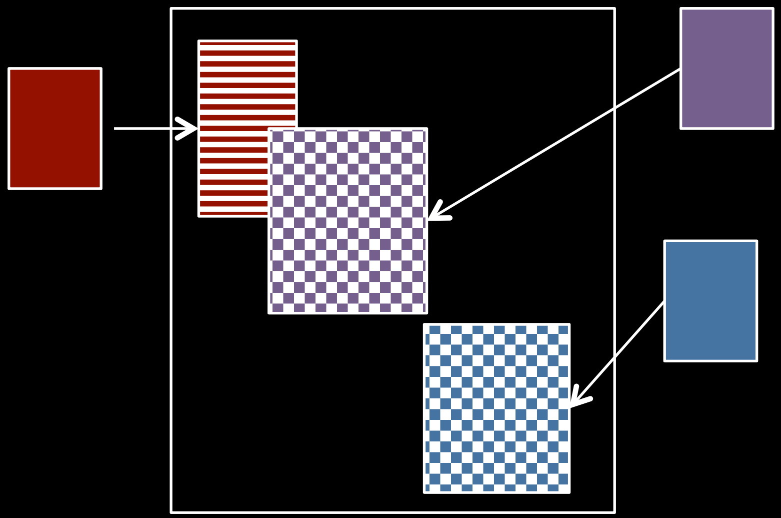
- *First-ever whole-mantle seismic model from numerical waveform tomography*
- *Finding: Most* volcanic hotspots are linked to two spots on the boundary between the metal core and rocky mantle 1,800 miles below Earth's surface.

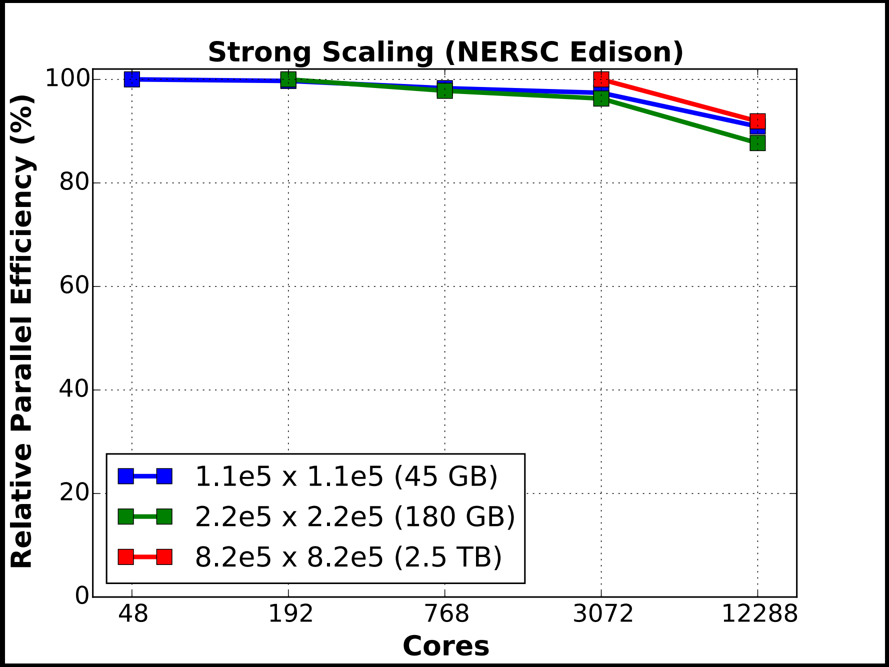
Makss unsolvable problems solvable!



Scott French, Barbara Romanowicz, *"Broad plumes rooted at the base of the Earth's mantle beneath major hotspots",* **Nature, 2015**

Data Fusion for Observation with Simulation

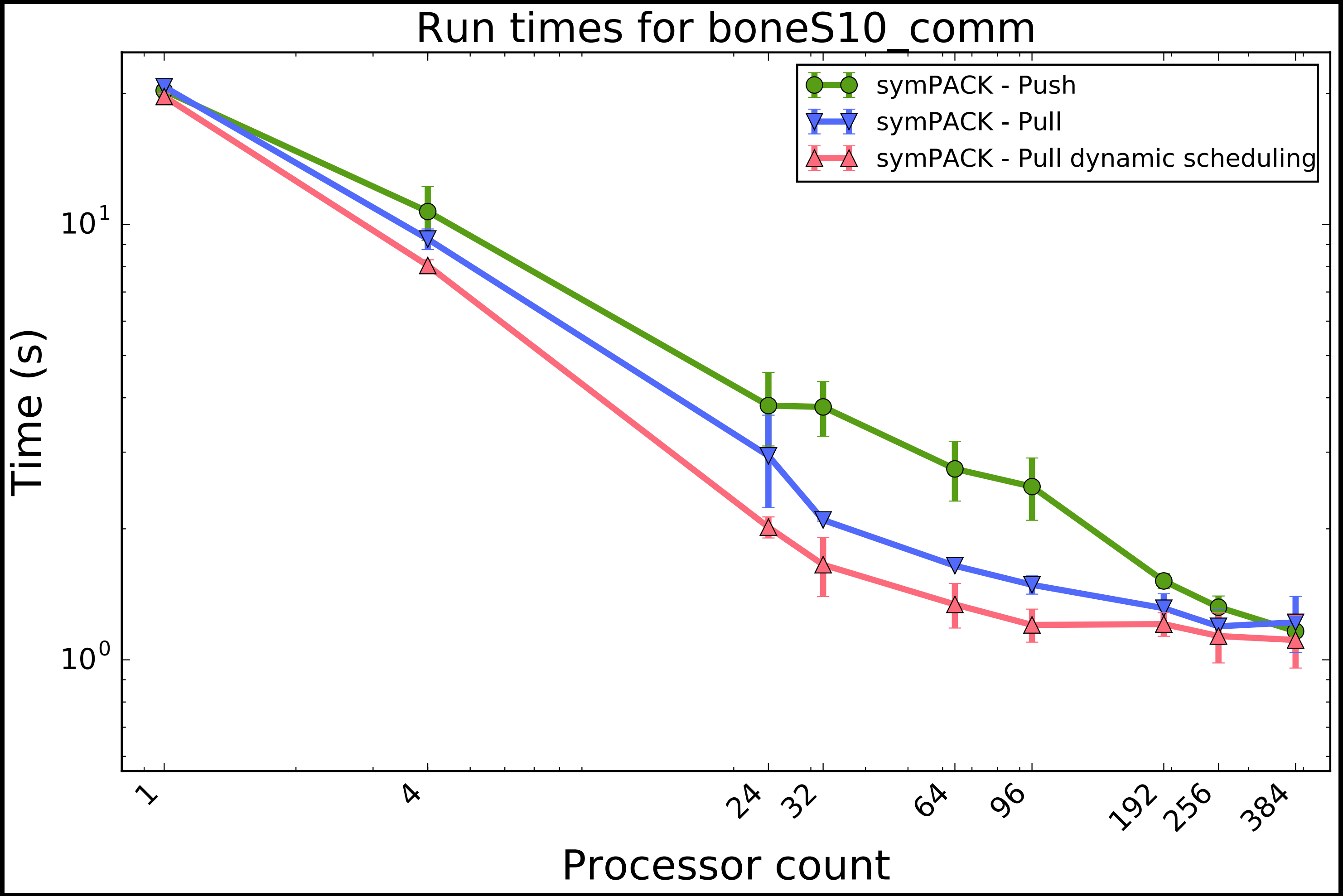




- **Unaligned data from observation**
- **One-sided strided updates**
- **Could MPI-3.0 one-sided do this? Yes, but not well so far**

Scott French, Y. Zheng, B. Romanowicz, K. Yelick

Sparse Cholesky in PGAS (UPC++)



Fan-both algorithm by Jacquelin & Ng, in UPC++

Unstructured, Graph-based, Data analytics problem: De novo Genome Assembly

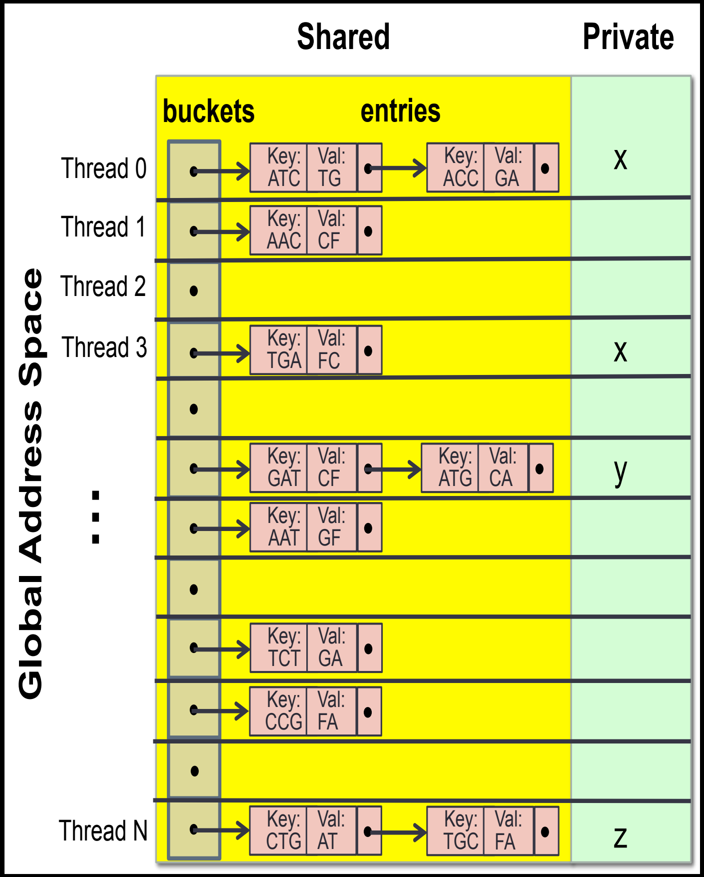
- **DNA sequence consists of 4 bases: A/C/G/T**
- **Read: short fragment of DNA sequence that can be read by a DNA sequencing technology – can't read whole DNA at once.**
- **De novo genome assembly: Reconstruct an unknown genome from a collection of short reads.**
  - Constructing a jigsaw puzzle without having the picture on the box

Random Access Graph Analytics

- Genome assembly "needs shared memory"

Global Address Space

- Low overhead communication
- Remote atomics
- Partitions for any structure



Scales to 15K+ cores

Under 10 minutes for human

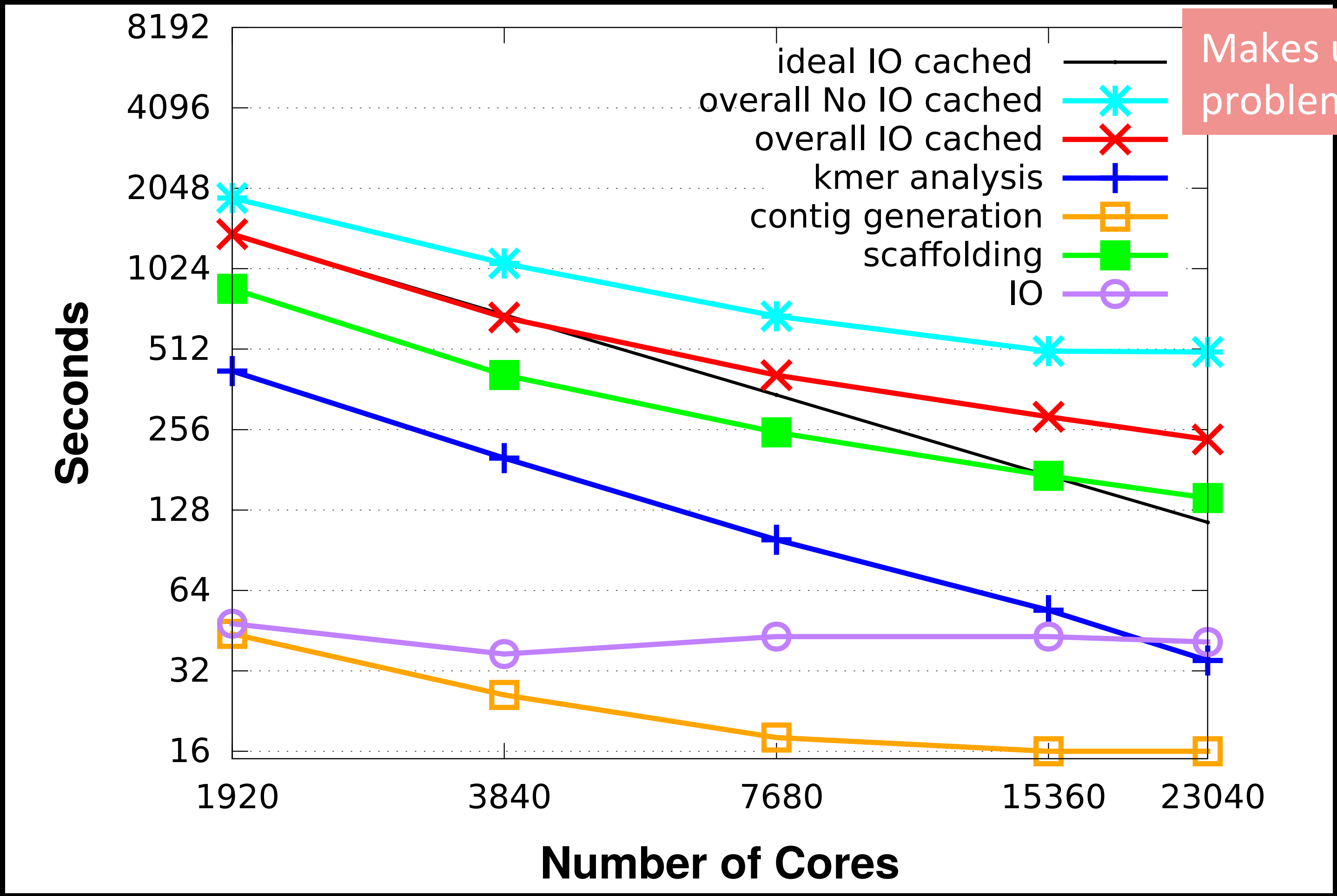
First ever solution

E. Georganas, A. Buluc, J. Chapman, S. Hofmeyr, C. Aluru, R. Egan, L. Oliker, D. Rokhsar, K. Yelick

Many types of distributed hash tables in HipMer

- **Global update only**
  - Can aggregate and reorder updates
- **Global read only**
  - Sometimes with good hash locality so caching helps
- **Global read-modify-write of elements in table**
  - Remote atomics
- **Local read and write**
  - Separate all-to-all or reduction phase

Strong scaling (human genome) on Cray XC30



Makes unsolvable problems solvable!

- Complete assembly of human genome in **4 minutes using 23K cores.**
- **700x speedup over** original Meraculous (took **2,880 minutes** on large shared memory with some Perl code); Some problems (wheat, squid, only run on HipMer version)

Roadmap for this talk

- ✓ **Science at the boundary of simulation and observation**
- ✓ **Science data challenges**
- ✓ **What do I mean by Exascale?**
- **Exascale challenges for Data problems**
  - ✓ Programming models
  - – Algorithms: Communication avoidance
  - – Architectures
  - – Systems
  - – Policies

Beyond Domain Decomposition

2.5D Matrix Multiply on BG/P, 16K nodes / 64K cores

Surprises:

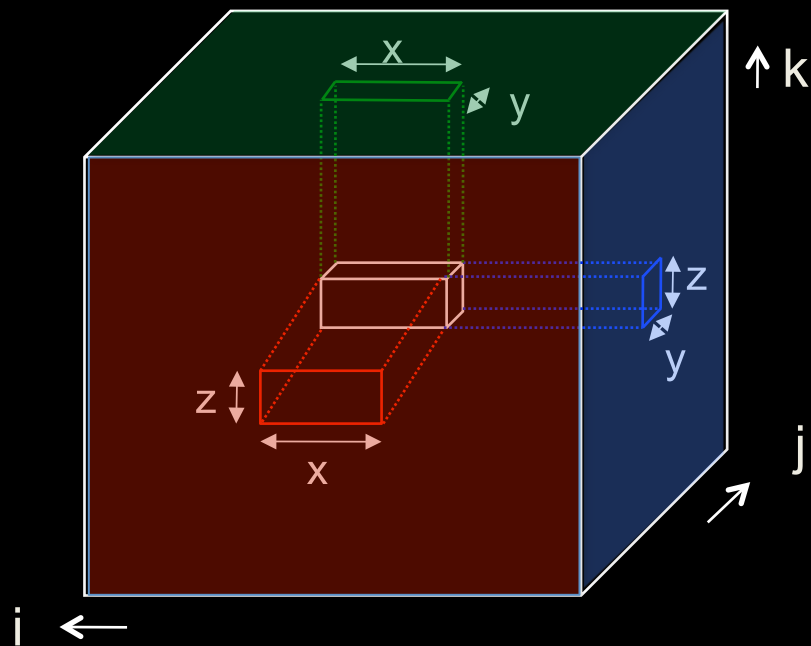
- Even Matrix Multiply had room for improvement
- Idea: make copies of C matrix (as in prior 3D algorithm, but not as many)
- Result is provably optimal in communication

Lesson: Never waste fast memory
And don't get hung up on the owner computes rule

Can we generalize for compiler writers?

Deconstructing 2.5D Matrix Multiply

Solomonick & Demmel



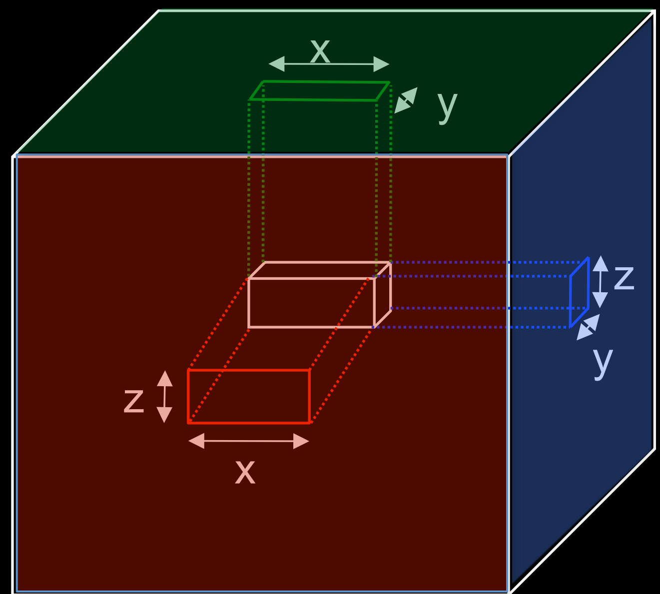
- **Tiling the iteration space**
- **2D algorithm: never chop k dim**
- **2.5 or 3D: Assume + is associative; chop k, which is → replication of C matrix**

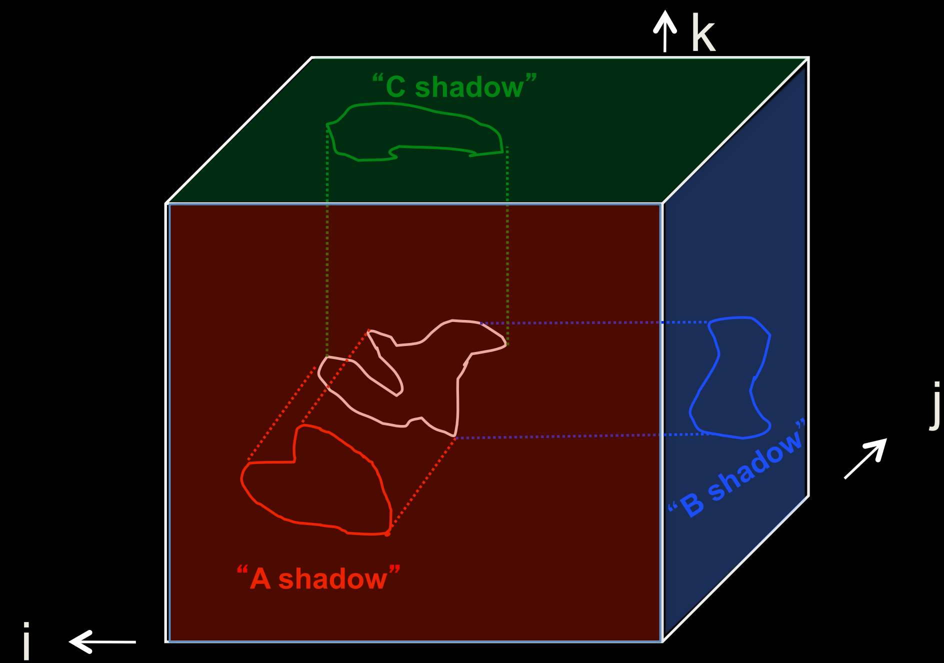
Matrix Multiplication code has a 3D iteration space
Each point in the space is a constant computation (*/+)

```
for i
  for j
    for k
        C[i,j] …  A[i,k] …  B[k,j]  …
```

Lower Bound Idea on C = A*B

Iromy, Toledo, Tiskin





Cubes in black box with
side lengths x, y and z
= Volume of black box
= x*y*z
= (#A□s * #B□s * #C□s)$^{1/2}$
= (xz * zy * yx)$^{1/2}$

(i,k) is in "A shadow" if (i,j,k) in 3D set
(j,k) is in "B shadow" if (i,j,k) in 3D set
(i,j) is in "C shadow" if (i,j,k) in 3D set

Thm (Loomis & Whitney, 1949)
 # cubes in 3D set = Volume of 3D set
≤ (area(A shadow) * area(B shadow) * area(C shadow)) $^{1/2}$

Generalizing Communication Lower Bounds and Optimal Algorithms

- **For serial matmul, we know #words_moved = $\Omega(n^3/M^{1/2})$, attained by tile sizes $M^{1/2}$ x $M^{1/2}$**

- **Thm (Christ,Demmel,Knight,Scanlon,Yelick):** *For any program that "smells like" nested loops, accessing arrays with subscripts that are linear functions of the loop indices*

  #words_moved = Ω (#iterations/M^e)

  for some e we can determine
- **Thm (C/D/K/S/Y): Under some assumptions, we can determine the optimal tiles sizes**
  - E.g., index expressions are just subsets of indices
- **Long term goal: All compilers should generate communication optimal code from nested loops**

Implications for Compilers

x += …

x += …

x += …

x += …

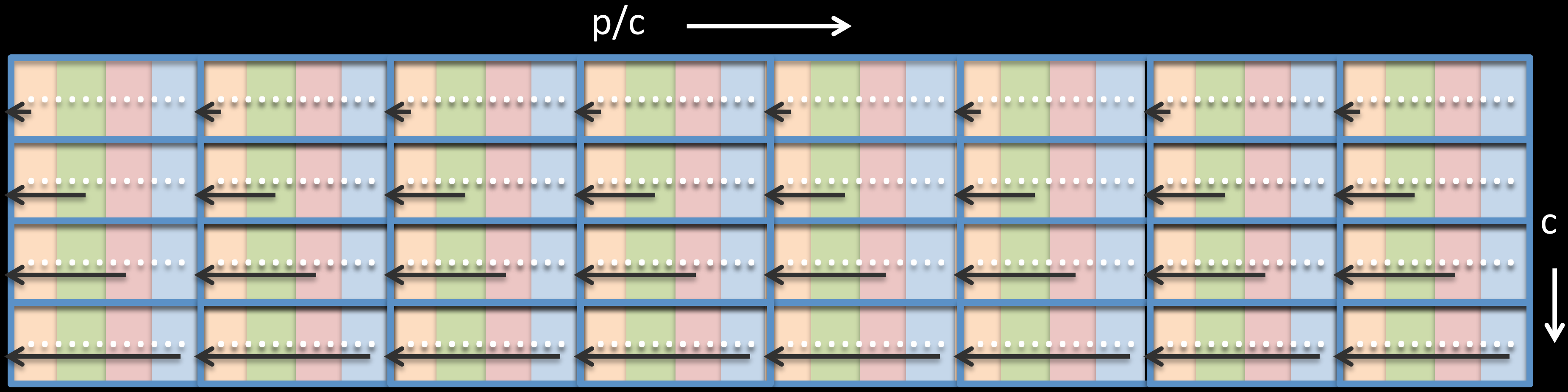
Using x for C[i,j] here

- **Much of the work on compilers is based on owner-computes**
  - For MM: Divide C into chunks, schedule movement of A/B
  - Data-driven domain decomposition partitions data; but we can partition work instead
- **Ways to compute C "pencil"**
  1. Serially
  2. Parallel reduction
  3. Parallel asynchronous (atomic) updates
  4. Or any hybrid of these

Standard vectorization trick

- **For what types / operators does this work?**
  - "+" is associative for 1,2 rest of RHS is "simple"
  - and commutative for 3

Communication Avoiding Version (using a "1.5D" decomposition)



- **Divide p into c groups. Replicate particles within group.**
  - First row responsible for updating all by orange, second all by green,…
- **Algorithm: shift copy of n/(p*c) particles to the left**
  - Combine with previous data before passing further level (log steps)
- **Reduce across c to produce final value for each particle**
- Total Computation: $O(n^2/p)$;
- Total Communication: O(log(p/c) + log c) messages,

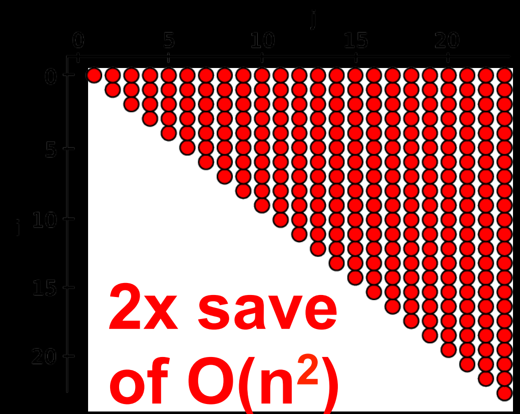
  O(n*(c/p+1/c)) words

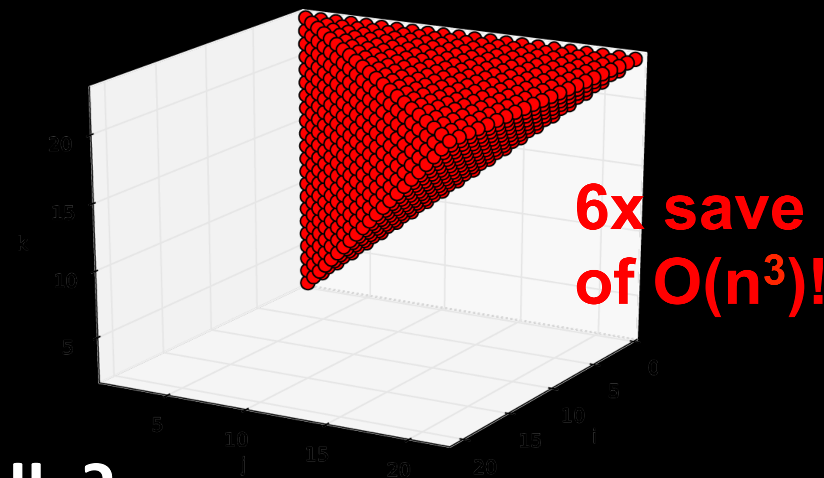
Limit: $c \leq p^{1/2}$

Driscoll, Georganas, Koanantakool, Solomonik, Yelick

Challenge: Symmetry & Load Balance

- Force symmetry ($f_{ij} = -f_{ji}$) saves computation
- 2-body force matrix vs 3-body force cube



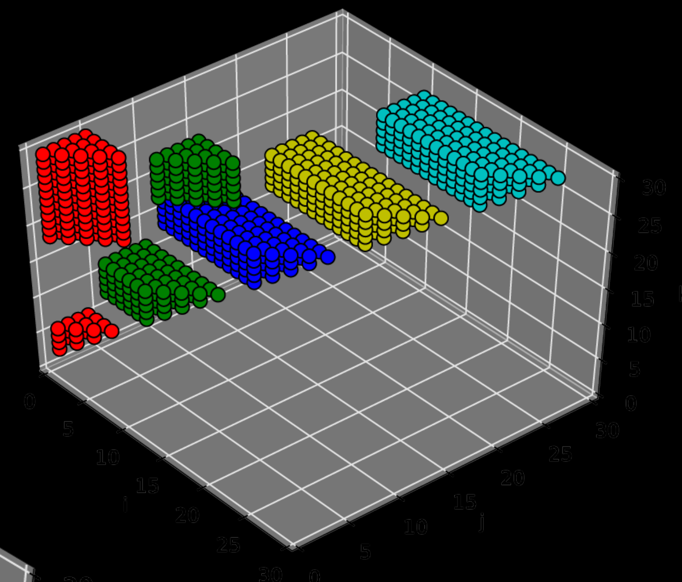


- How to divide work equally?

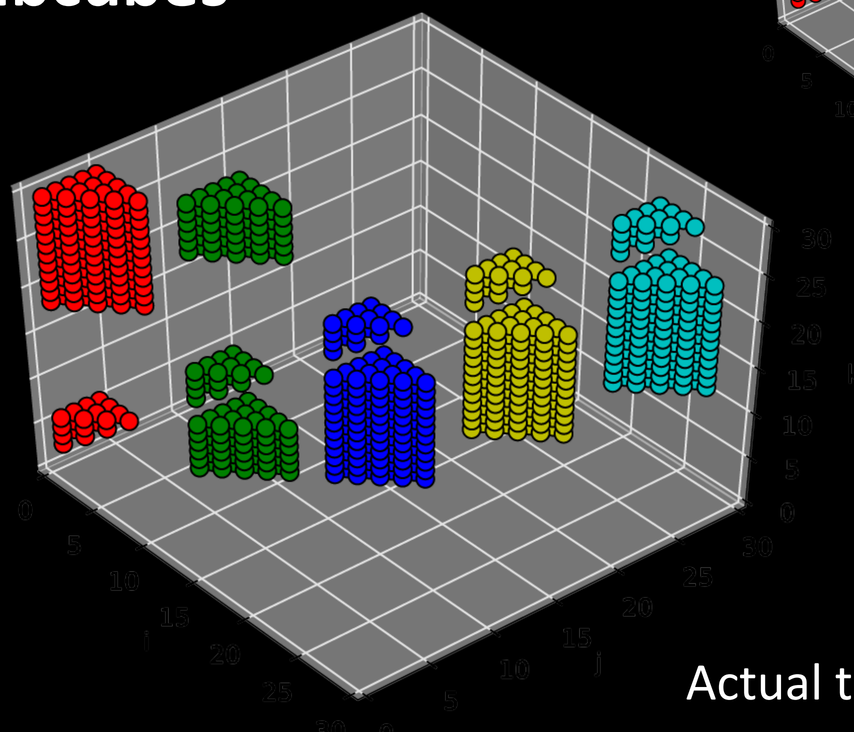
Koanantakool & Yelick

3-Way N-Body Animation

- p=5, n=30
- 6 particles per processor
- 5x5 subcubes



Equivalent triplets in the big tetrahedron

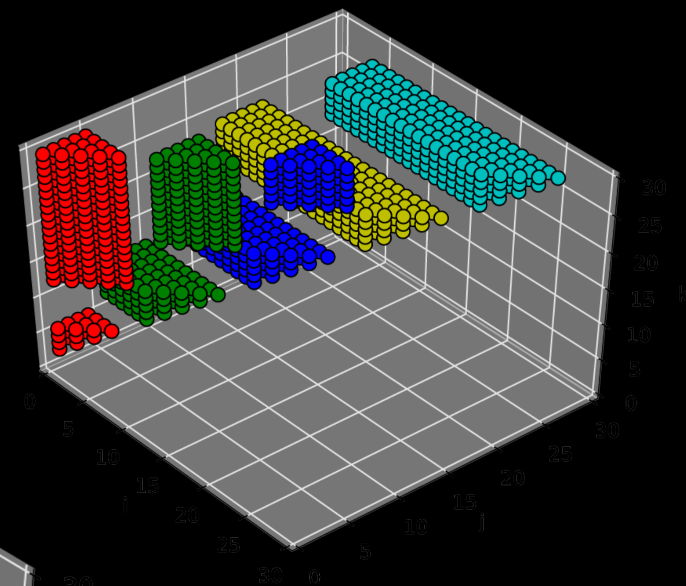


Actual triplets

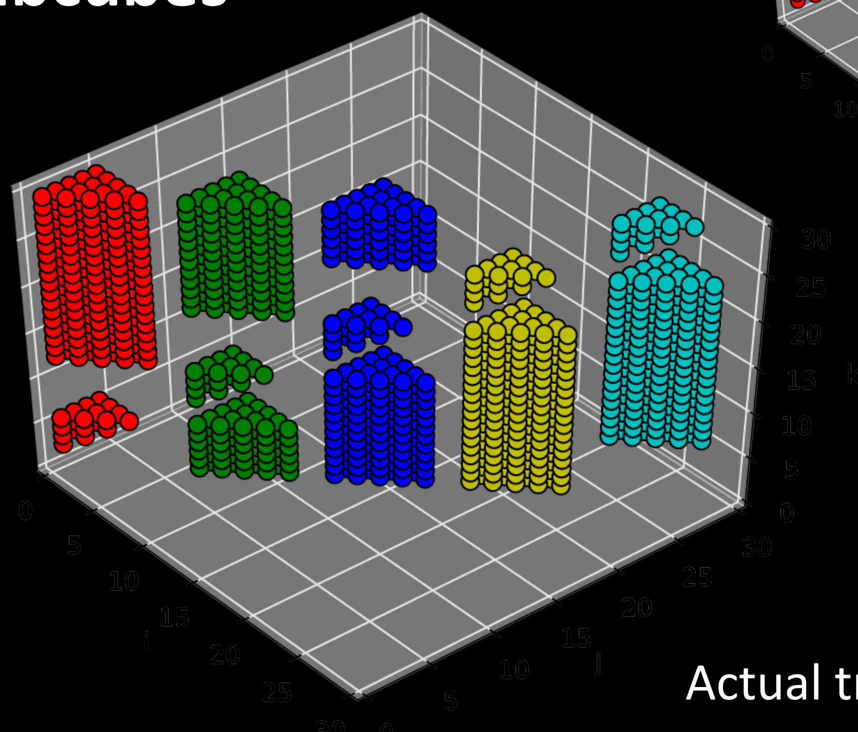
Koanantakool & Yelick

3-Way N-Body Animation

- p=5, n=30
- 6 particles per processor
- 5x5 subcubes



Equivalent triplets in the big tetrahedron

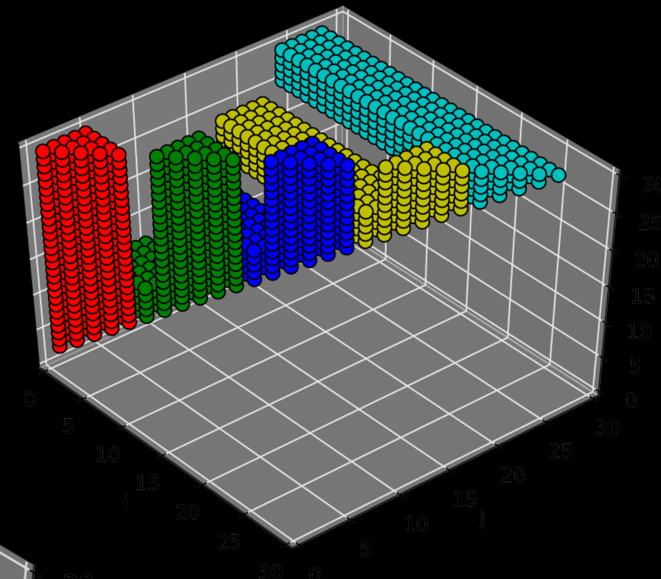


Actual triplets

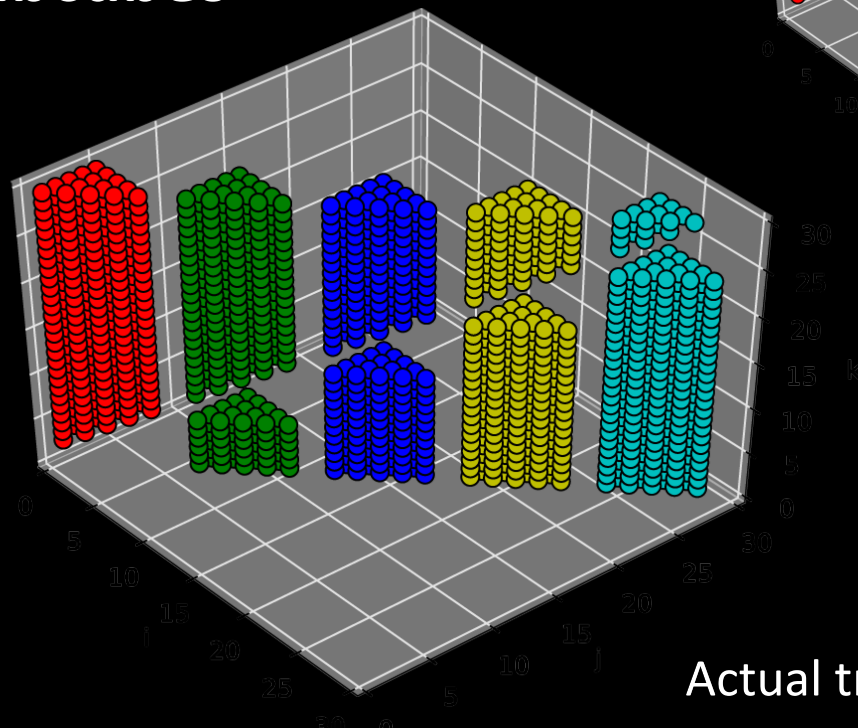
Koanantakool & Yelick

3-Way N-Body Animation

- p=5, n=30
- 6 particles per processor
- 5x5 subcubes



Equivalent triplets in the big tetrahedron

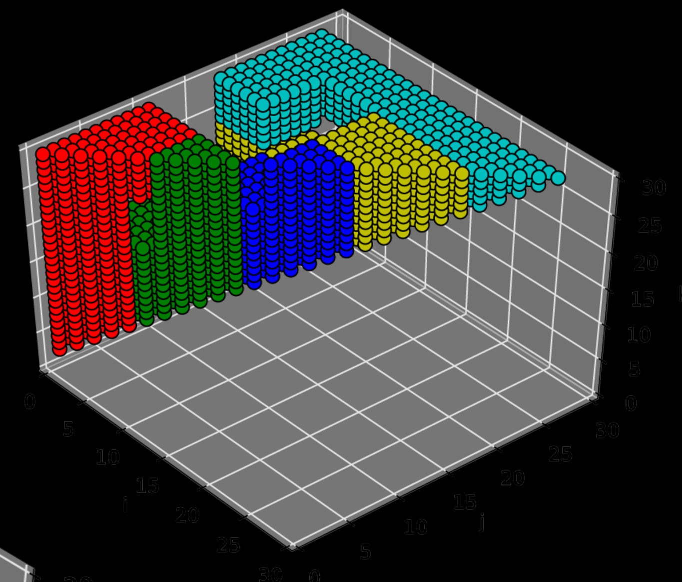


Actual triplets

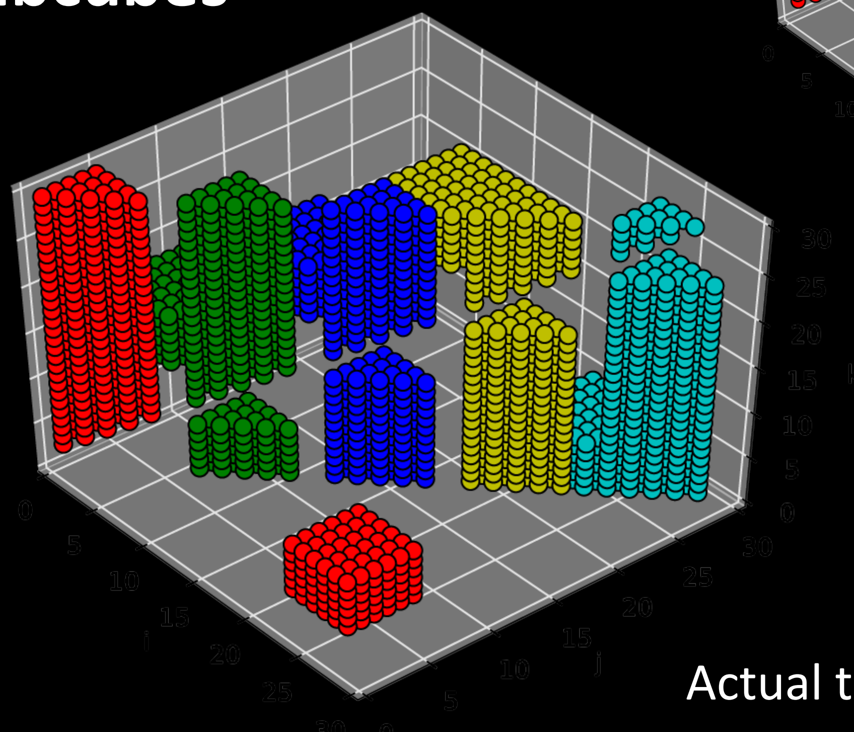
Koanantakool & Yelick

3-Way N-Body Animation

- p=5, n=30
- 6 particles per processor
- 5x5 subcubes



Equivalent triplets in the big tetrahedron

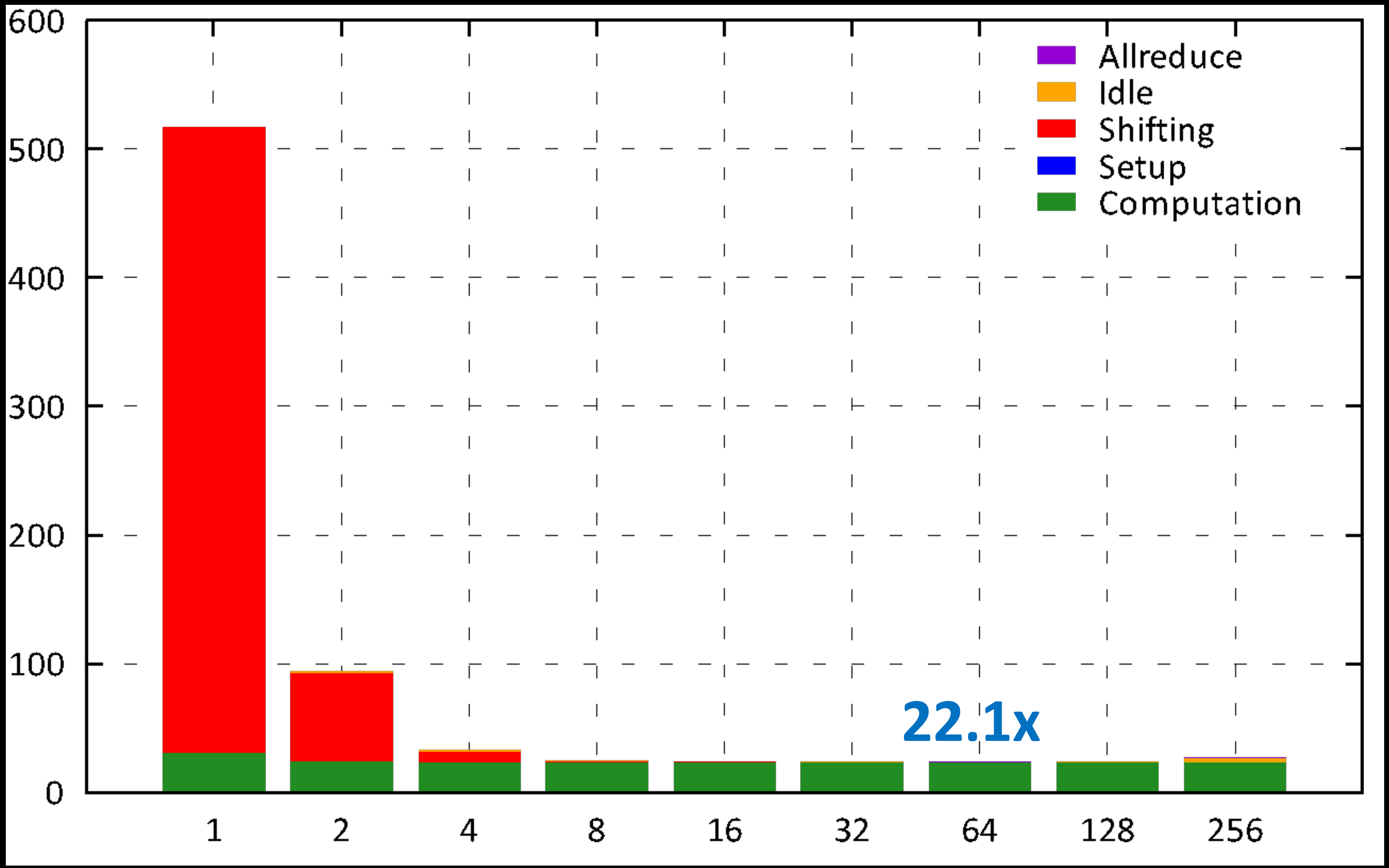


Actual triplets

Koanantakool & Yelick

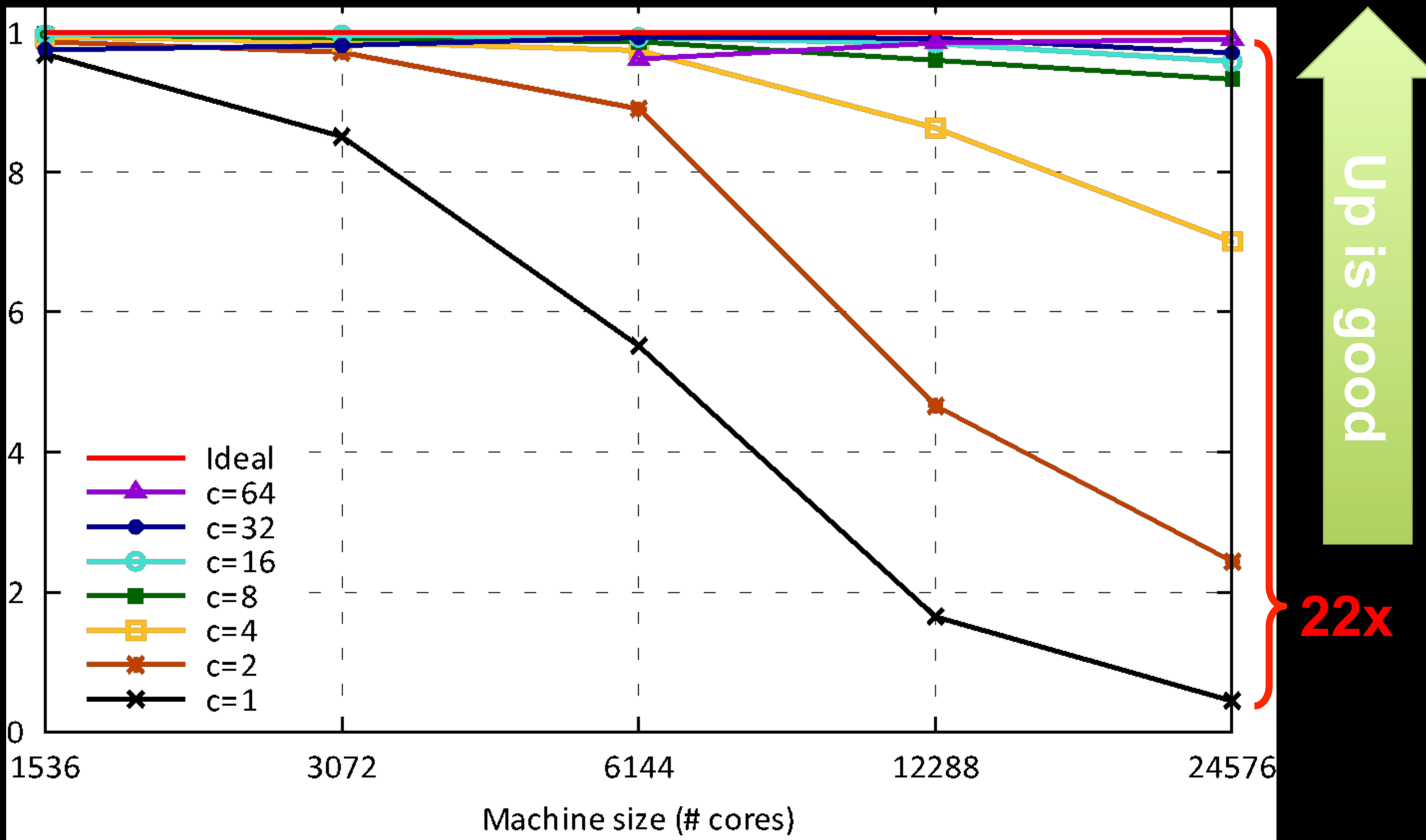
3-Way N-Body Speedup

- Cray XC30, 24k cores, 24k particles



Koanantakool & Yelick

Strong Scaling of .5D Algorithns

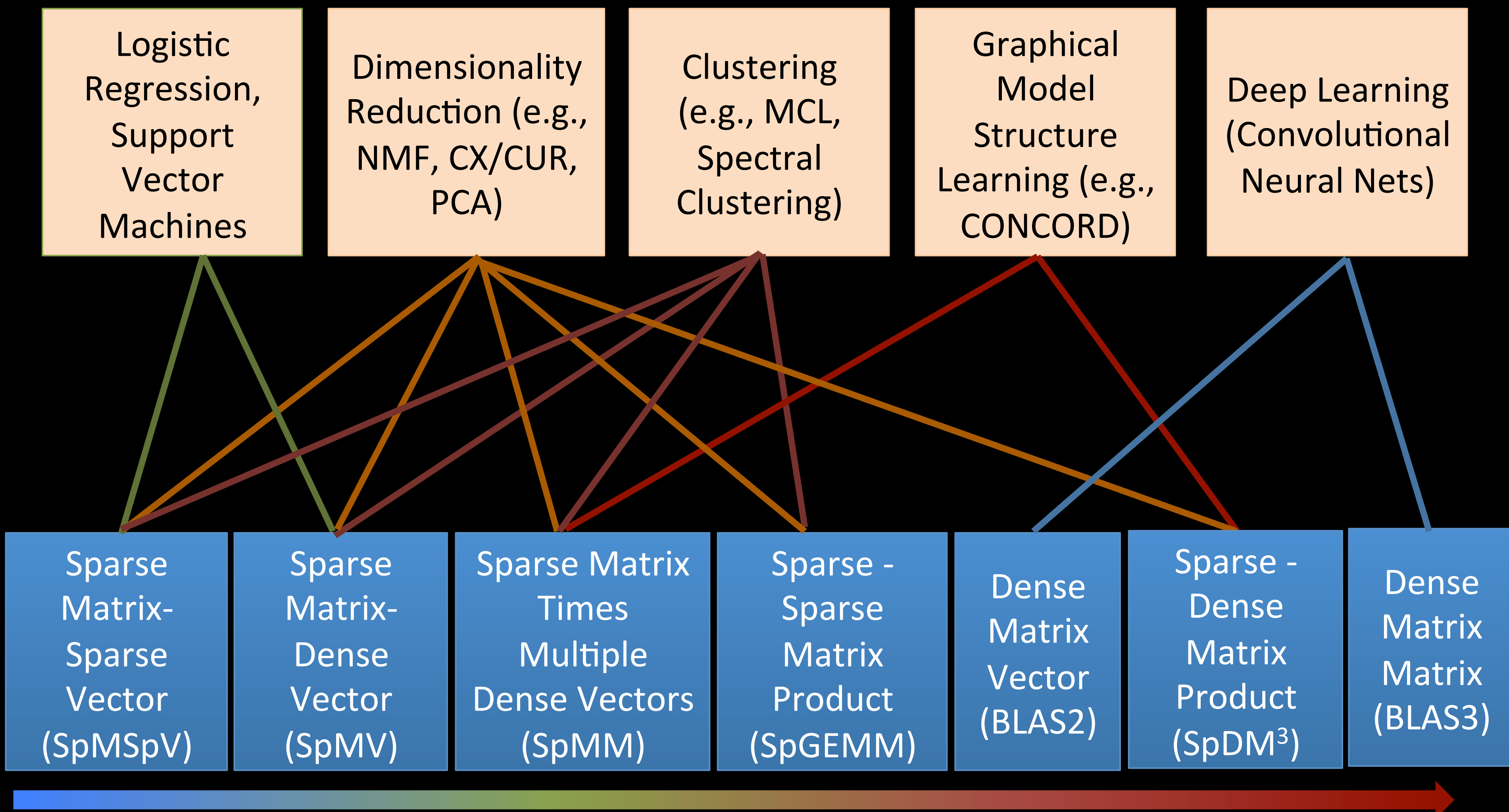


Koanantakool & Yelick

Analytics vs. Simulation Kernels:

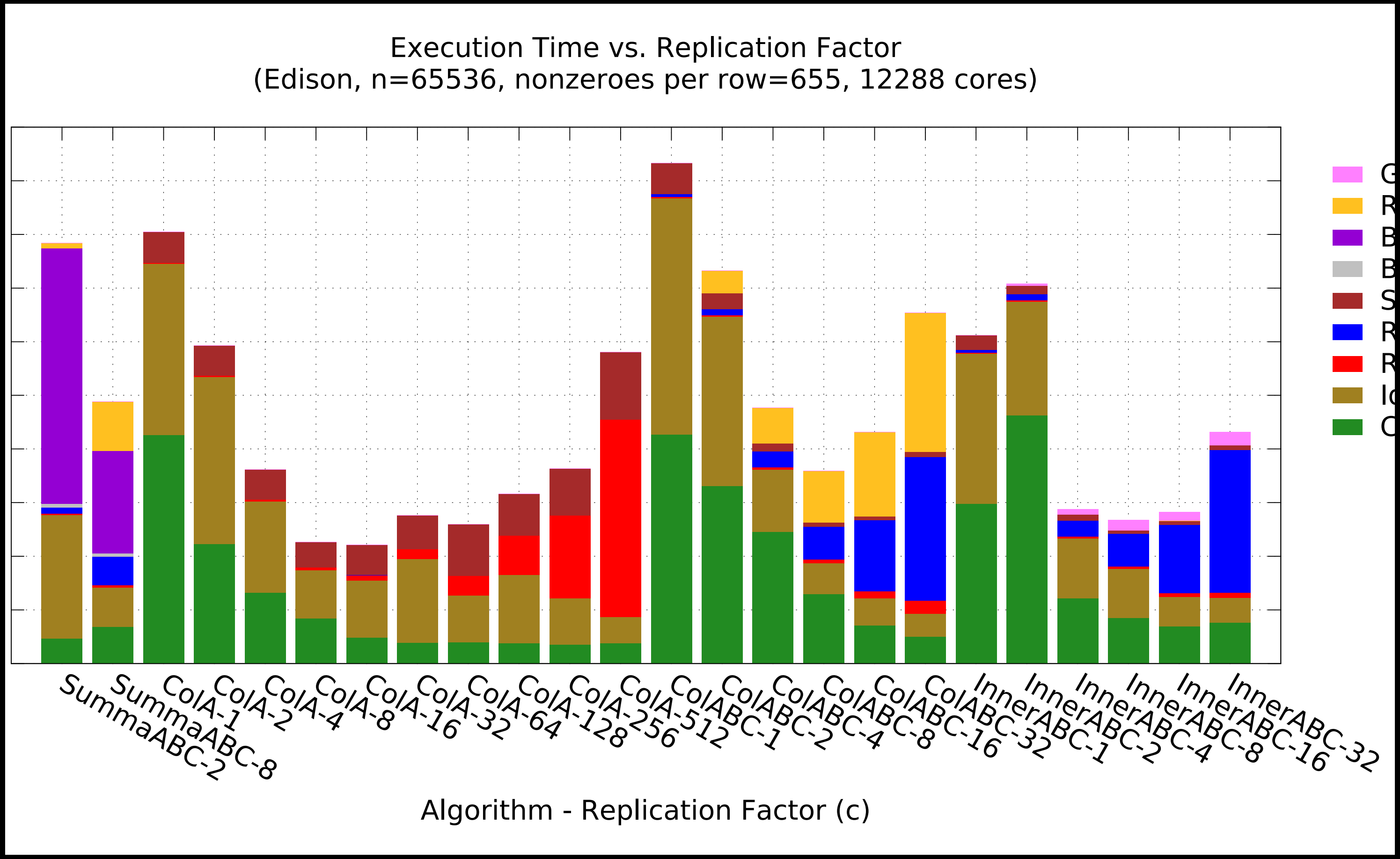
7 Giants of Data	7 Dwarfs of Simulation
Basic statistics	Monte Carlo methods
Generalized N-Body	Particle methods
Graph-theory	Unstructured meshes
Linear algebra	Dense Linear Algebra
Optimizations	Sparse Linear Algebra
Integrations	Spectral methods
Alignment	Structured Meshes

Machine Learning Mapping to Linear Algebra



Aydin Buluc

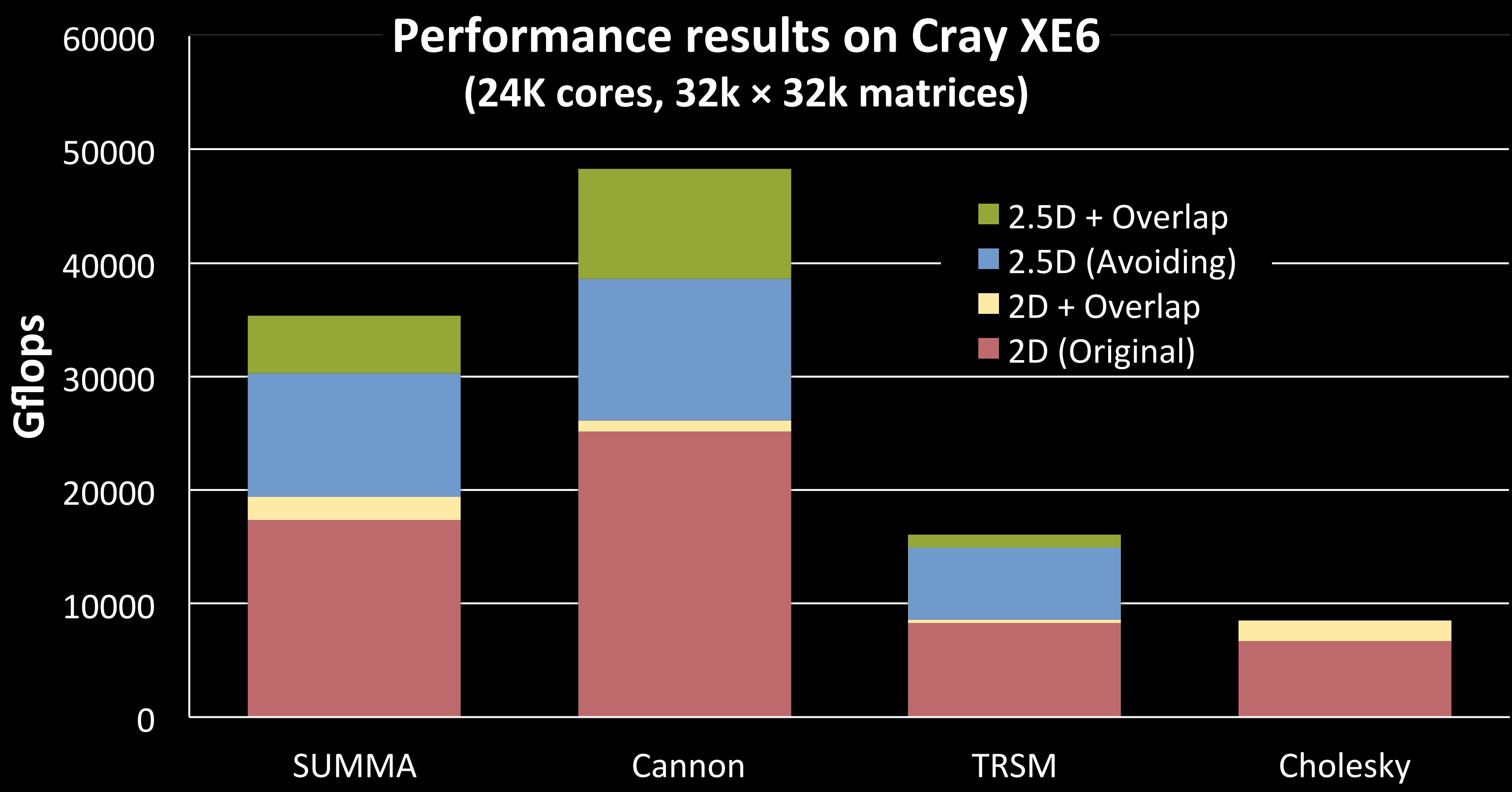
Sparse-Dense Matrix Multiply Too!



- **Variety of algorithms that divide in or 2 dimensions**

Koanantakool & Yelick

Communication Overlap Complements Avoidance



Even with communication-optimal algorithms (minimized bandwidth) there are still benefits to overlap and other things that speed up networks

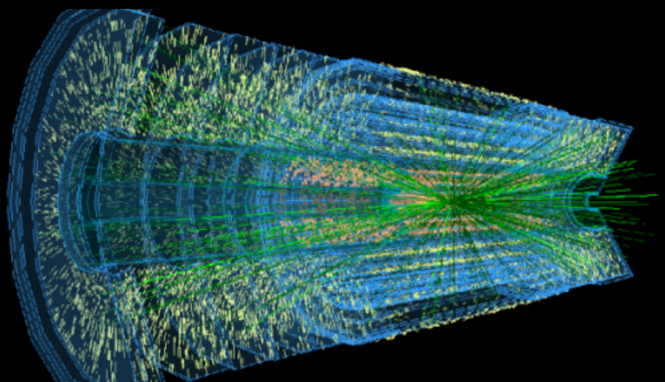
SC'12 paper (Georganas, González-Domínguez, Solomonik, Zheng, Touriño, Yelick)

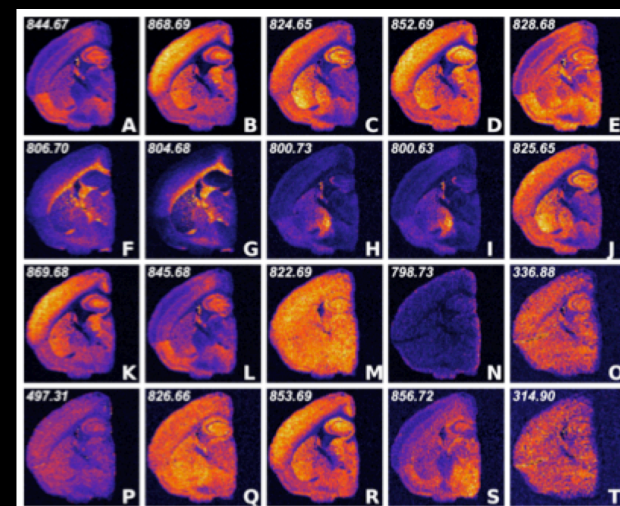
Roadmap for this talk

- ✓ **Science at the boundary of simulation and observation**
- ✓ **Science data challenges**
- ✓ **What do I mean by Exascale?**
- **Exascale challenges for Data problems**
  - ✓ Programming models
  - ✓ Algorithms
  - – Architectures
  - – Systems
  - – Policies

Data processing with special purpose hardware

- General trend towards specialization for continued performance growth
- Data processing (on raw data) will be first in science





Particle Tracking with Neuromorphic chips

Deep learning processors for image analysis

Computing in Detectors

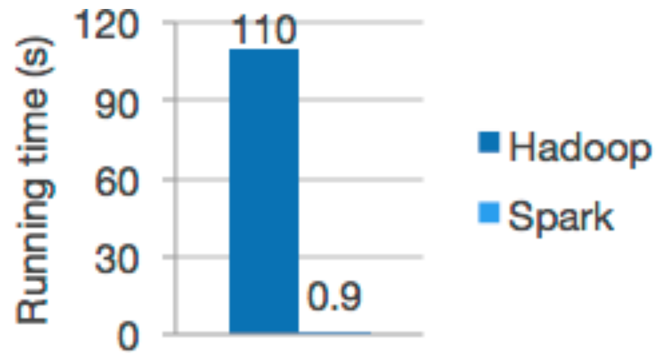
FPGAS for genome analysis

And can we also use these for simulation?

Productive Programming

Speed

Run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk.



- High failure rate
- Slow network
- Fast (local) disk

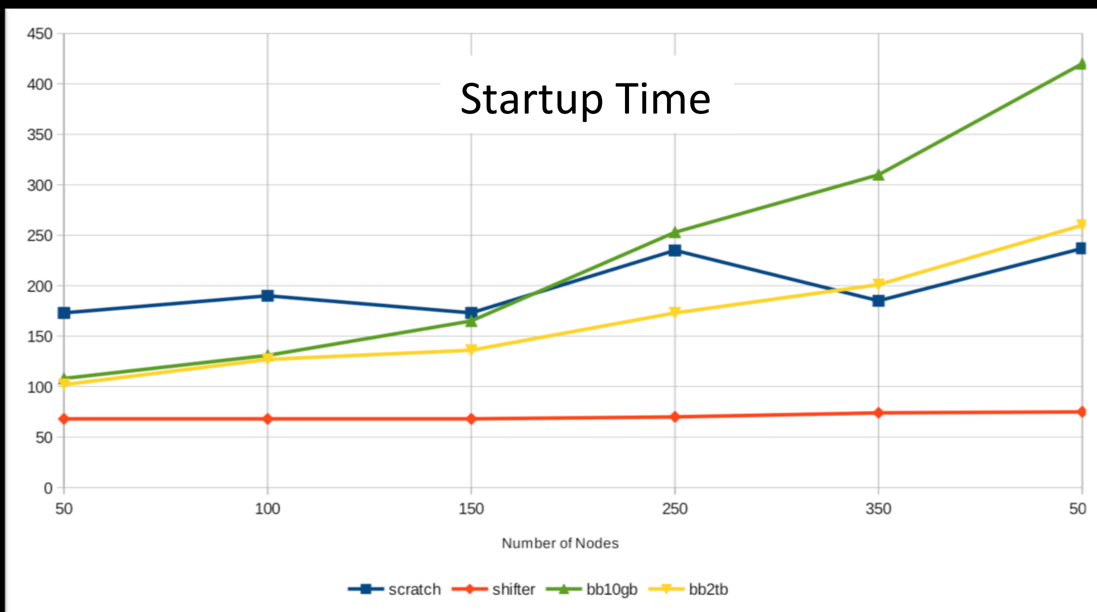
And Spark is still 10x+ slower than MPI

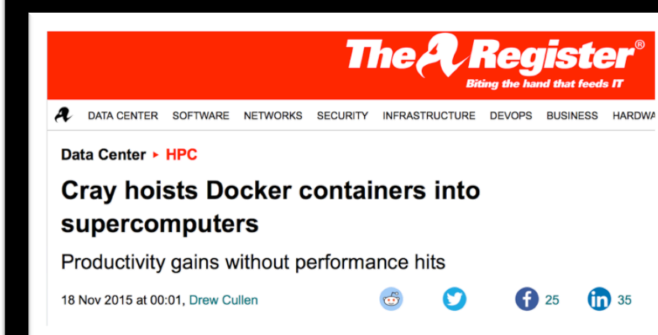
Systems configured for data-intensive science

NERSC Cori has data partition (Phase 1, Haswell) pre-exascale (Phase 2, KNL preproduction)
WAN-to-Cori optimized for streaming data: 100x faster from LCLS to Cori and Globus to CERN

Containers for HPC Systems

- Data analysis pipelines are often large, complex software stacks
- NERSC Shifter (with Cray), supports containers for HPC systems
- Used in HEP and NP projects (ATLAS, ALICE, STAR, LSST, DESI)




The Register
Biting the hand that feeds IT
DATA CENTER SOFTWARE NETWORKS SECURITY INFRASTRUCTURE DEVOPS BUSINESS HARDWA

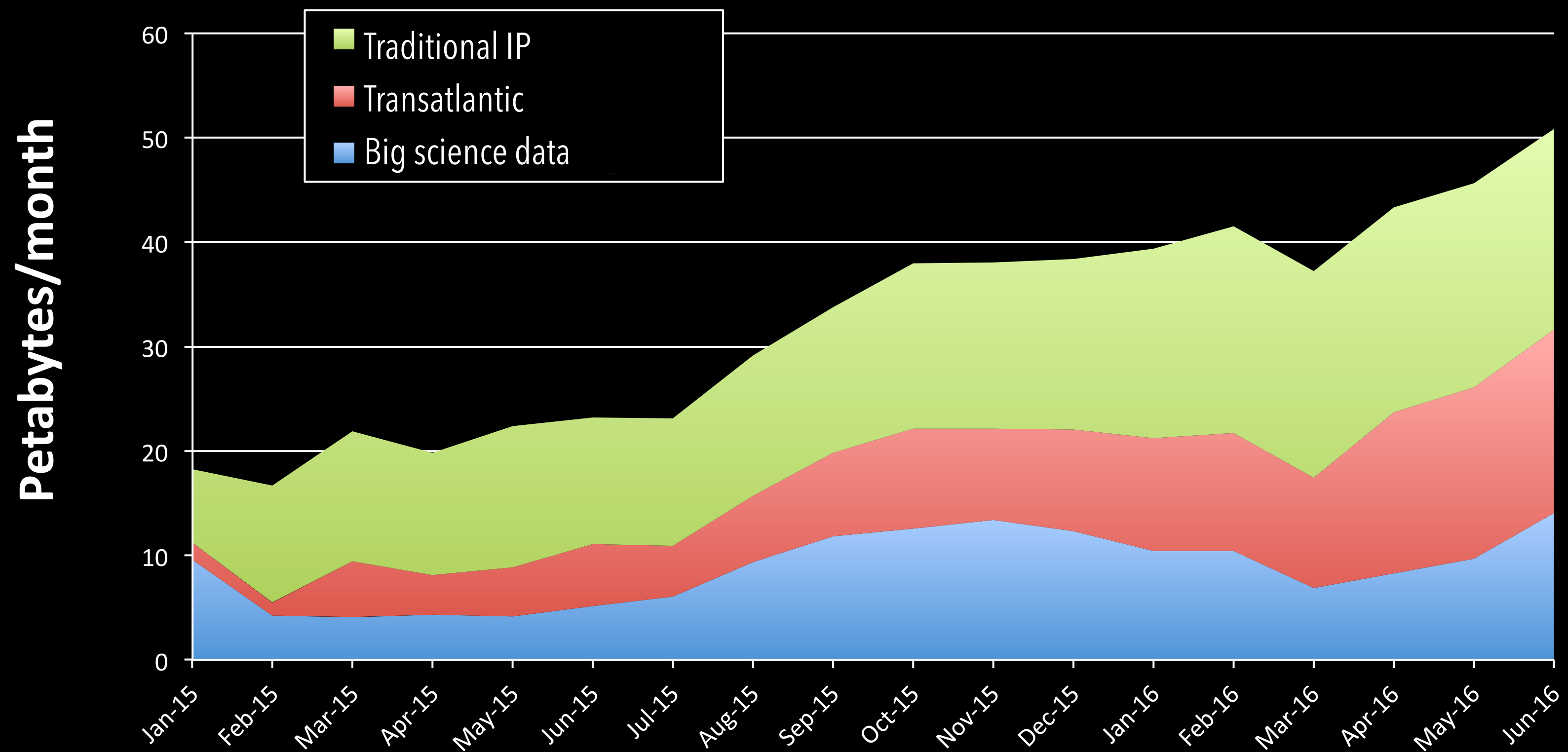
Data Center ▸ HPC

Cray hoists Docker containers into supercomputers

Productivity gains without performance hits

18 Nov 2015 at 00:01, Drew Cullen 25 35

ESnet: Exponential data growth drives capacity



Science DMZ to deliver bandwidth to the end users

OSCARS for bandwidth reservation

100 Exabytes/year by 2024!

Roadmap for this talk

- ✓ **Science at the boundary of simulation and observation**
- ✓ **Science data challenges**
- ✓ **What do I mean by Exascale?**
- **Exascale challenges for Data problems**
  - ✓ Programming models
  - ✓ Algorithms
  - ✓ Architectures
  - ✓ Systems
  - – Policies

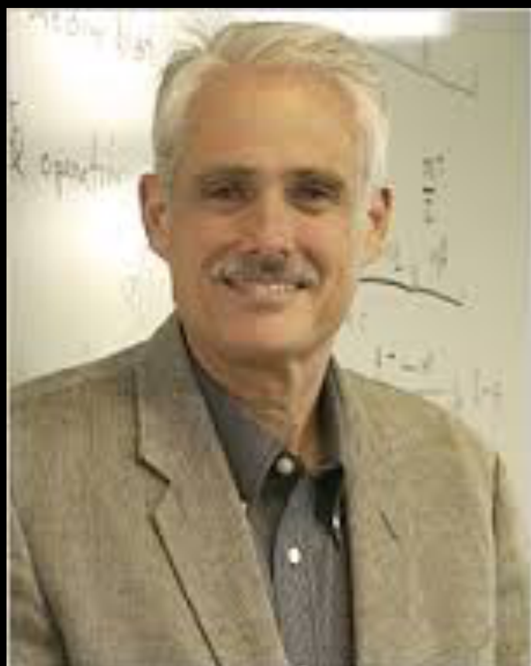
HPC Computing Policies

Cloud Computing Policies

Thanks to many collaborators

Eun-Jin Im Sam Williams Christian Bell
Arvind Krishnamurthy Bert de Jong
Jason Duell Carelton Miyamoto Nick Knight
Horst Simon Jimmy Su Greg Balls Ed Givelberg
Dan Rokhsar Chih-Po Wen Khaled Ibrahim Wei Chen
Soumen Chakrabarti Lauren Smith Kaushik Datta
Leonid Oliker Costin Iancu Ngeci Bowman Hongzhang Shan
Edgar Solomonik Jeff Jones Jim Demmel Yili Zheng Siu Man Yau
Shoaib Kamil Paul Hargrove Bob Lucas Etienne Deprit
Amir Kamil Bill Carlson Eric Roman Noah Treuhaft
Dan Bonachea C.J. Lin Tarek El-Ghazawi
Rajesh Nishtala Erich Strohmaier Sabrina Merchant
Mani Narayanan Randi Thomas Rich Vuduc
Harsha Simhadri Deborah Weisser Penporn Kaonantakool
Jarrod Chapman Steve Steinberg Evangelos Georganas
Brian Kazian Michael Driscoll David Ozog

And to Ken Kennedy



"Take care of your students and the rest will take care of itself."