

More Data, More Science and... Moore's Law

Kathy Yelick

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

Three parts

- **Science opportunities**
- **Technology challenges**
- **Productivity approaches**

Part I

Science is poised for transformation

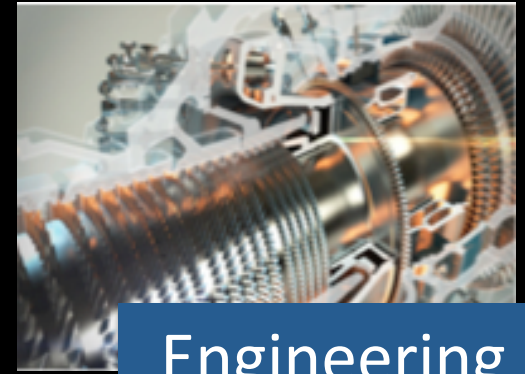
and Science in the broadest sense



Medicine



Basic Science



Engineering



Environment



Security

Experimental Science is Changing

By using our website you agree to our use of cookies in accordance with our cookie policy.

OK

PRIVACY POLICY +

JAX[®] MICE & SERVICES



JAX[®] Mice are the highest quality and most-published mouse models in the world. Take advantage of our large inventories of common inbred strains and the convenience of having your breeding and drug efficacy needs met by the leading experts in mouse modeling.

Search for Mice

Advanced Mice Search

Search for mice by strain, stock, gene, allele and synonyms

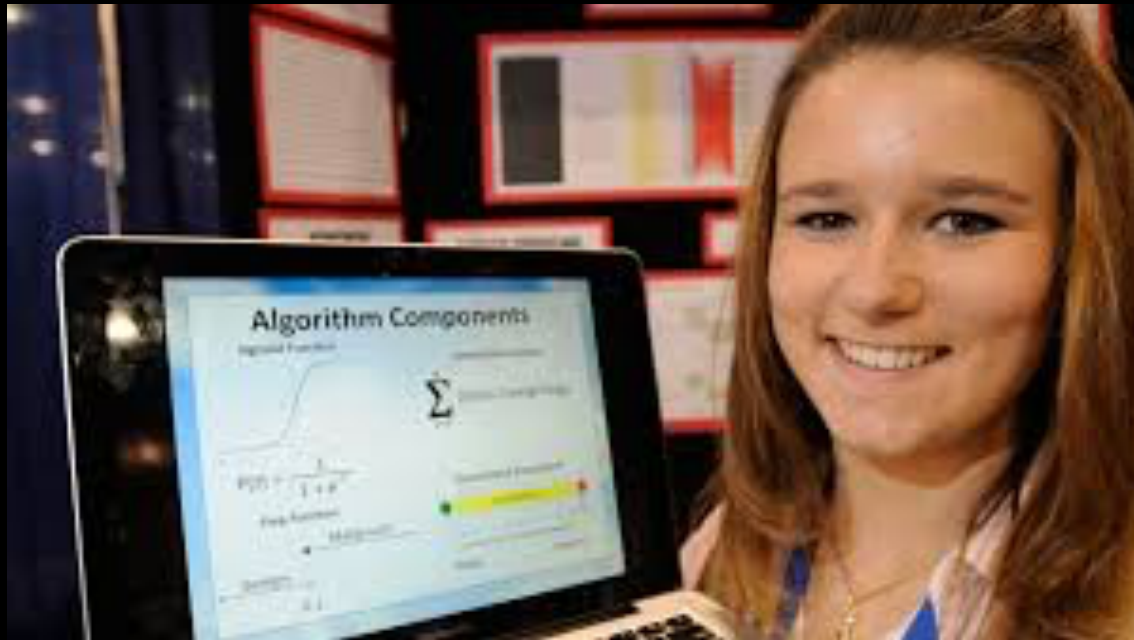


Breed Your Mouse

Test Your Drug

Cryopreserve Your Mouse

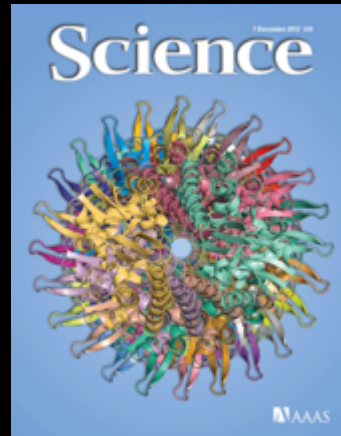
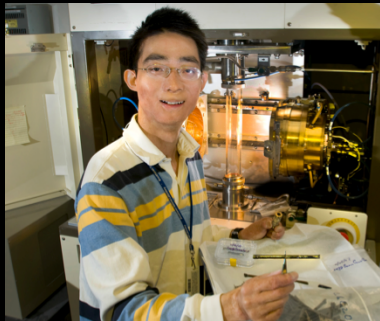
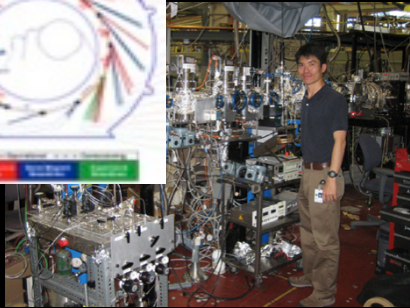
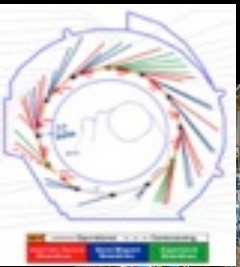
Empowering New Scientists



17-year-old Brittany Wegner creates breast cancer detection tool that is 99% accurate on a minimally invasive, previously inaccurate test.

Machine Learning + Online Data + Cloud Computing

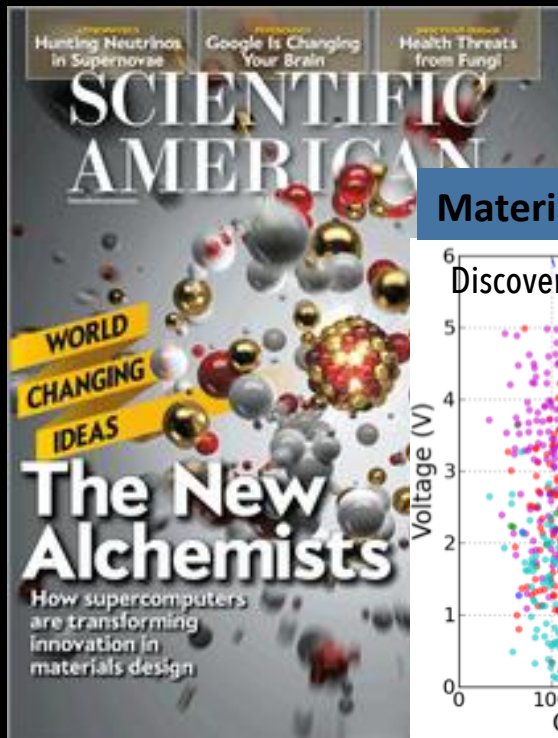
Old School Experimental Science



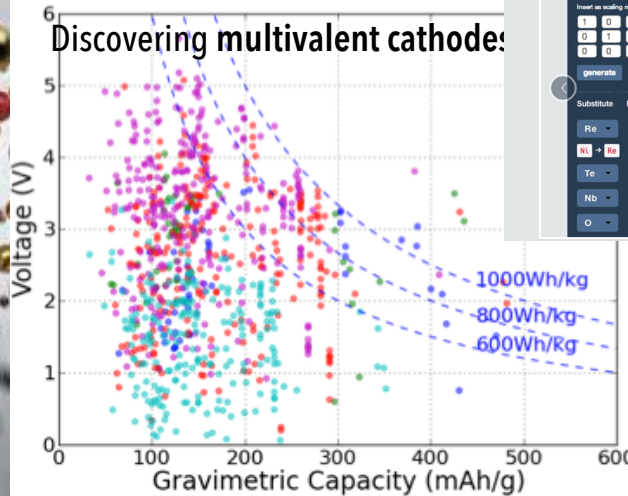
Computing, experiments, networking and expertise in a “Superfacility” for Science



Computational Scientists May Never Login



Materials Project



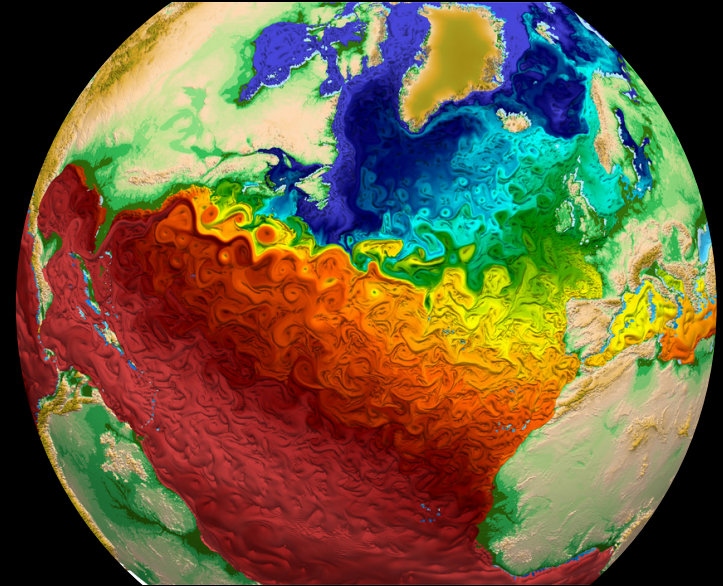
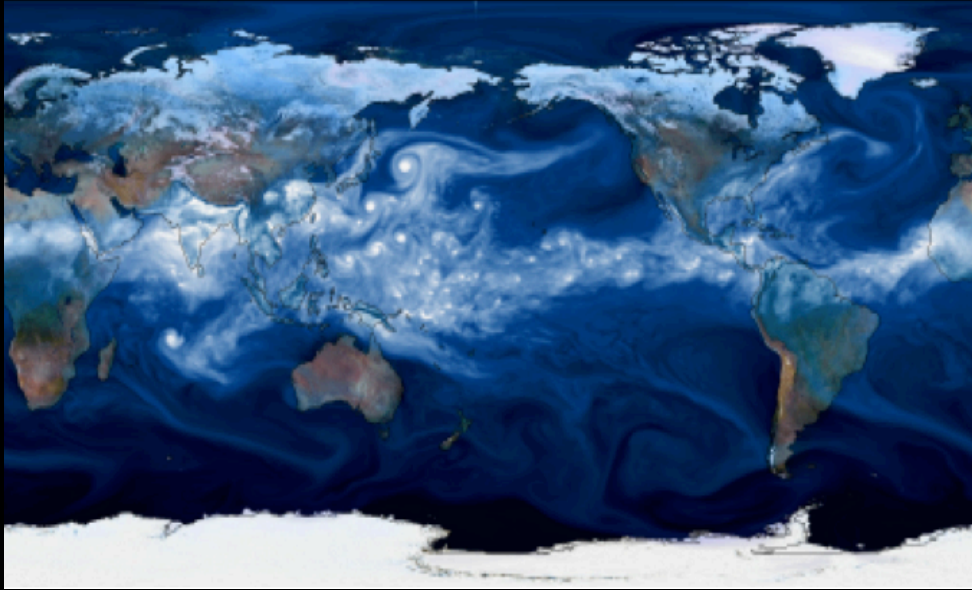
The Materials Project website header. It features the logo (three geometric shapes) and the text "The Materials Project". Below the logo is a paragraph: "Harnessing the power of supercomputing and state of the art electronic structure methods, the Materials Project provides open web-based access to computed information on known and predicted materials as well as powerful analysis tools to inspire and design novel materials." At the bottom are three buttons: "Learn more", "YouTube Tutorials", and "Sign In or Register to start using".

A screenshot of the Materials Project interface. On the left, "Supercell Dimensions" shows a 3x3 grid of cells with "generate" and "Substitute" buttons. The "Substitute" section lists Re, Ni, Te, Nb, and O. In the center is a 3D ball-and-stick model of a crystal structure. On the right, "Lattice Parameters" shows a, b, c axes (4.019 Å, 4.019 Å, 5.271 Å) and angles alpha, beta, gamma (all 90.000°). Below that, "Fractional Coordinates" shows a table for the "Re" atom with coordinates (0, 0, 0) and (0, 0, 0.5).

Novel computing technology here?

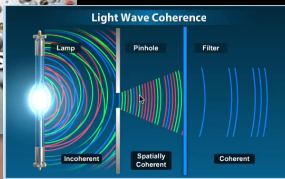
- Over 10,000 users of the Materials Project since 2011
- 1000x more computing needed for more materials, environments, and details

Simulations both consume and produce massive data

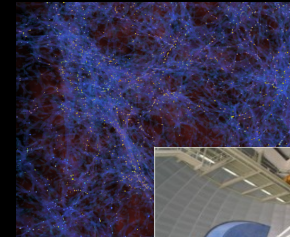


-
- **Simulation:** Resolve clouds, identify tipping points, quantify confidence, quantify agriculture and economics
 - **Observation:** Use environmental sensors and samples to refine models
 - **Analysis:** Machine learning identifies extreme events; Reanalysis of 100+ years of weather to improve
-

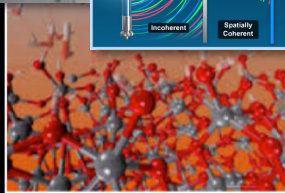
Breakthrough science will occur at the interface of observation and simulation



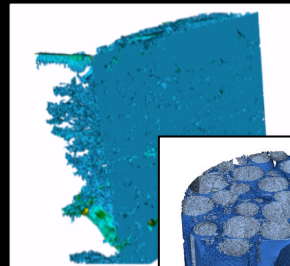
Environment and Climate



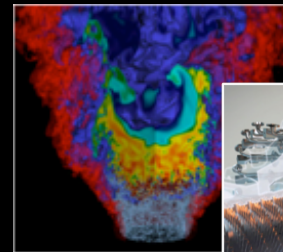
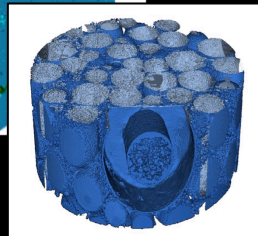
Cosmology



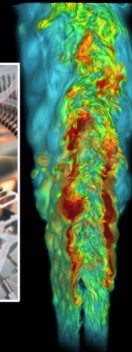
Materials and Chemistry



Subsurface science



Combustion



And it will need capable exascale systems, computer science, statistical machine learning and new mathematical models and algorithms

Challenge: Finding Scientific Data

Safari File Edit View History Bookmarks Window Help

www.google.com/search?tbs=sbi:AMhZZIu-Ft1o4xXIjhVjclUv_1GtY_1M9gV_1hy

Berkeley Lab (...) TeamSnap :: M... Google CalMail - You... Search Results...

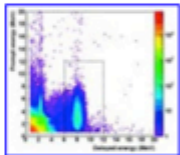
+You Search Images Mail Drive Calendar Sites Groups More -

CalMail - You must be logged in to a page.

Google Antineutrinos.jpg

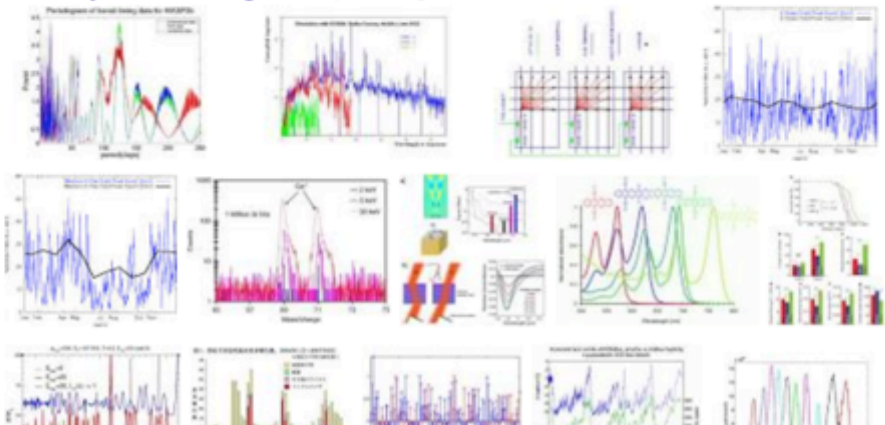
Web **Images** Maps Shopping More Search tools

Tip: Try entering a descriptive word in the search box.

 Image size:
153 × 133

No other sizes of this image found.

[Visually similar images](#) - Report images



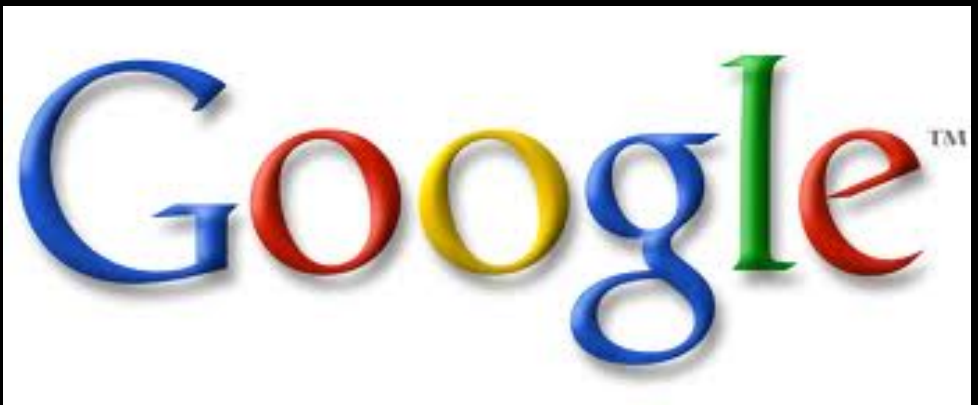
Challenge: Filtering, De-Noiseing and Curating Data



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

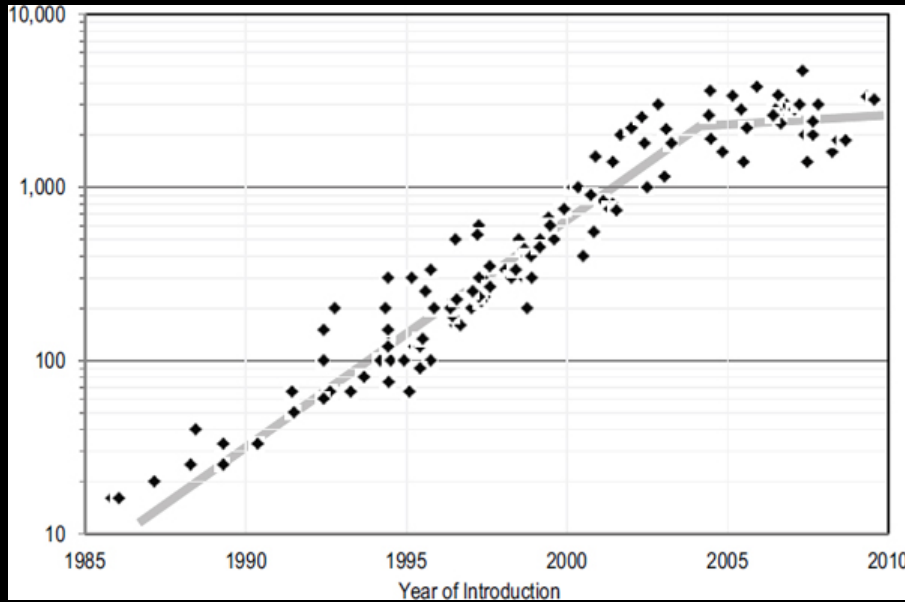
The Black Swans of Computing with 1990s Technology



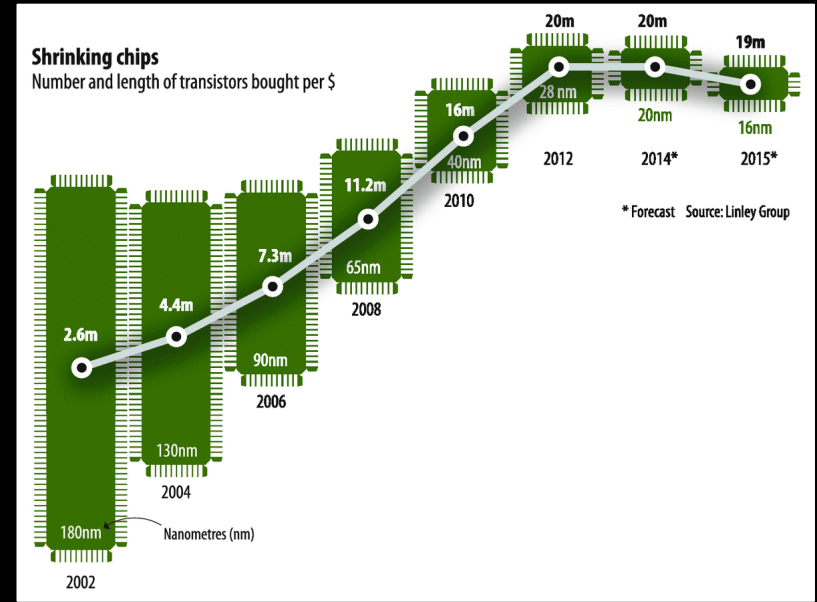
Part II

- **Science is poised for transformation**
- **Computing performance growth is slowing**

Computing Performance Stalls



2004: End of clock speed scaling



202?: End of transistor density scaling

We need an end game for Moore's Law and a new game after that

A Generic 2020+ Node Architecture

Memory Stacks on Package

Low Capacity, High Bandwidth, Software Control?

Lightweight Cores
(tiny, simple, massively parallel)
Throughput-Optimized

Bulky Cores

Latency Optimized

DRAM

DRAM

DRAM

DRAM

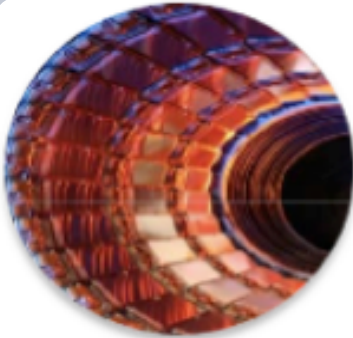
NVRAM: Burst
Buffers / rack-local
storage (software
control)

Part III

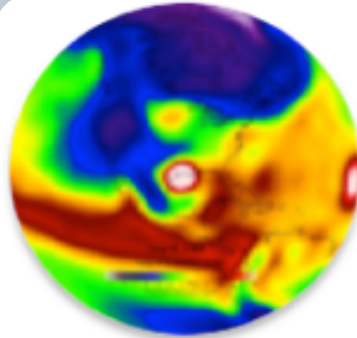
- **Science is poised for transformation**
- **Computing performance growth is slowing**
- **How more productive tools can help**

(What productivity means to me)

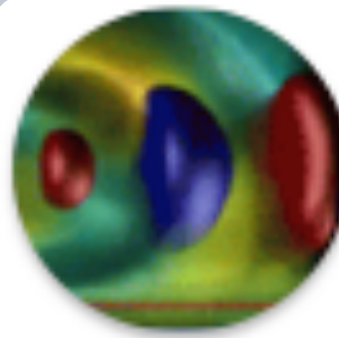
The right machine for the problem



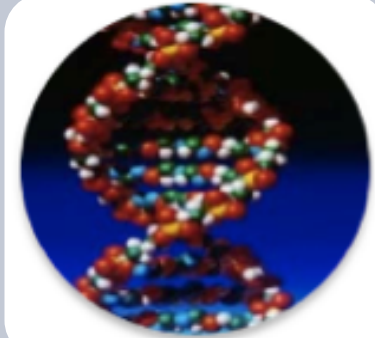
Massive
Independent
Jobs for
Analysis and
Simulations



Nearest
Neighbor
Simulations



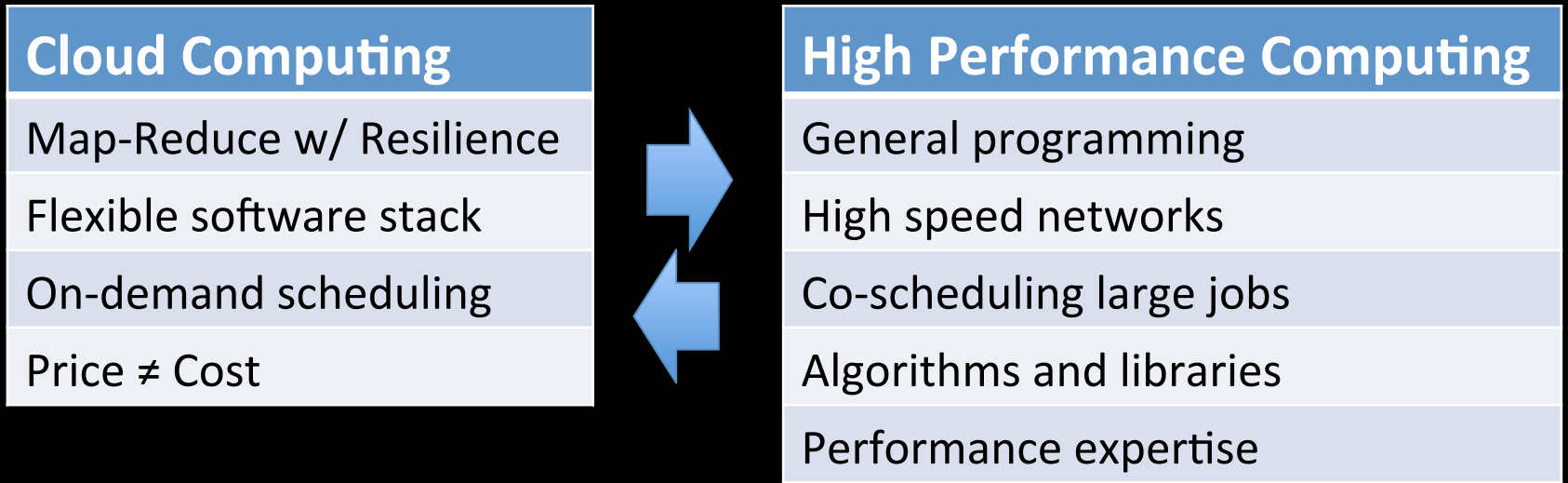
All-to-All
Simulations



Random
access, large
data Analysis

Applications often fit in multiple categories

The right systems software for the problem



Ideas and technologies have transferred in both directions

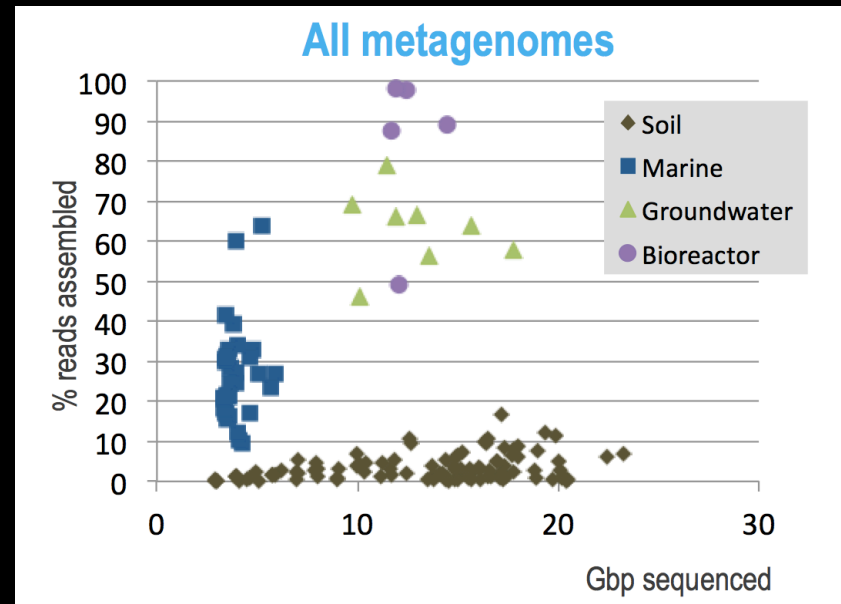
Ultimately, cloud computing is business model not a technology

A convenient programming model for the problem



- De Novo Genome assembly for the environment and health
“Scaffolding” step on shared memory
- Human: 44 hours to 20 secs
 - Wheat: “doesn’t run” to 32 secs

A programming model (UPC) that encourages the right algorithmic thinking (hash tables, atomics,...)



All 2015 human sequences could have been assembled on 10% of annual NERSC resources; metagenomes are much harder

A Network that Makes Location Irrelevant



Technology, policy, and social challenges across:

- Networks and countries
- Facilities
- Universities (Science DMZ)
- Systems (SDN)

ESnet Goal: Scientific progress **completely unconstrained by the location** of people, instruments, computational resources, or data.

Use scientific libraries and frameworks

- *Encapsulate the best algorithms*
- *Hide machine details*

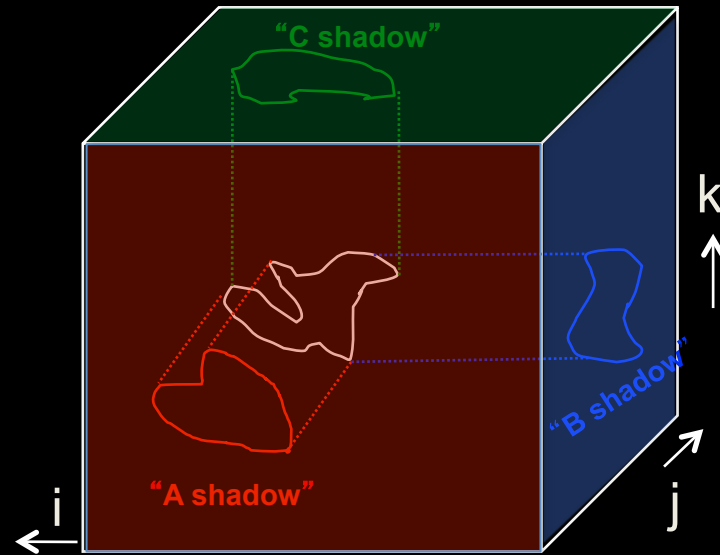
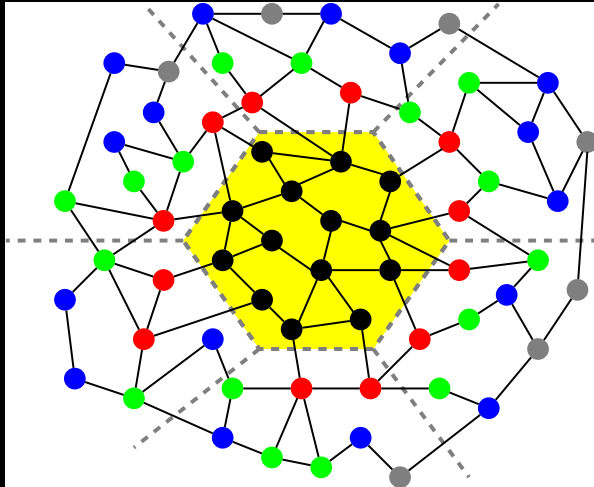


Enabled by portable programming models like MPI

The right libraries for the problem

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

The right algorithms, models and methods



Optimize for data movement as first-class concern

- Change numerical methods (e.g., high order)
- Design algorithms that minimize data movement

High level programming abstractions



Where is Performance Portability?

- **Multiple generations of GPUs, Xeon Phi, ...**
 - Not performance portable across systems
- **How do applications address this problem?**

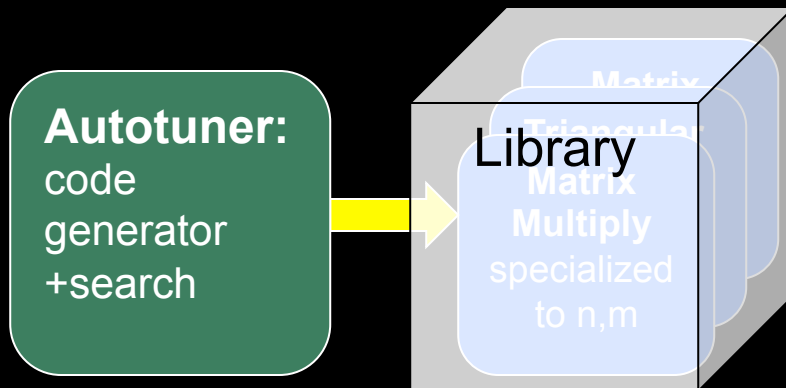
`#ifdef`

Approach 1: Use of libraries, templates and preprocessors

Approach 2: OpenMP4 absorbs necessary accelerator features (likely), but code still diverges

Approach 3: Architectures converge “enough” by 2023, perhaps with co-design help

Write code generators and autotuners



Two unsolved compiler problems

- Dependence analysis
- Optimization selection

What code generators do we have?

Dense Linear Algebra	Atlas
Spectral Algorithms	FFTW, Spiral
Particle	TCE
Sparse Linear Algebra	OSKI
Structured grids	TBD
Unstructured Grids	
Monte Carlo	

Develop and influence languages and compilers

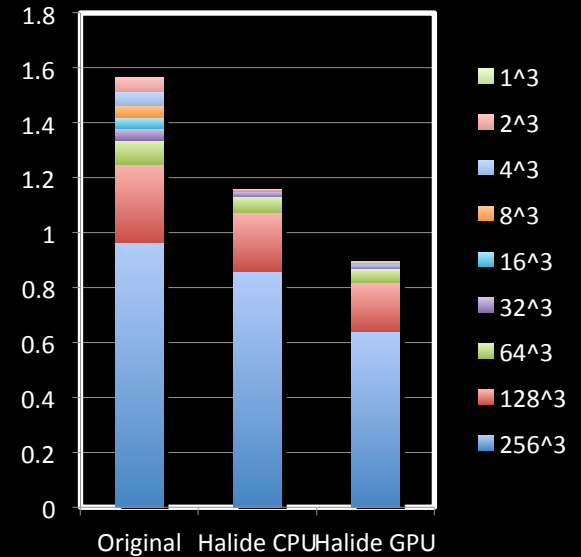
Rose
compiler
input

```
graph TD
    subgraph Rose_Compiler_Input [Rose compiler input]
        direction TB
        I1[Code block 1]
        I2[Code block 2]
        I3[Code block 3]
        I4[Code block 4]
        I5[Code block 5]
        I6[Code block 6]
        I7[Code block 7]
        I8[Code block 8]
        I9[Code block 9]
        I10[Code block 10]
        I11[Code block 11]
        I12[Code block 12]
        I13[Code block 13]
        I14[Code block 14]
        I15[Code block 15]
        I16[Code block 16]
        I17[Code block 17]
        I18[Code block 18]
        I19[Code block 19]
        I20[Code block 20]
        I21[Code block 21]
        I22[Code block 22]
        I23[Code block 23]
        I24[Code block 24]
        I25[Code block 25]
        I26[Code block 26]
        I27[Code block 27]
        I28[Code block 28]
        I29[Code block 29]
        I30[Code block 30]
        I31[Code block 31]
        I32[Code block 32]
        I33[Code block 33]
        I34[Code block 34]
        I35[Code block 35]
        I36[Code block 36]
        I37[Code block 37]
        I38[Code block 38]
        I39[Code block 39]
        I40[Code block 40]
        I41[Code block 41]
        I42[Code block 42]
        I43[Code block 43]
        I44[Code block 44]
        I45[Code block 45]
        I46[Code block 46]
        I47[Code block 47]
        I48[Code block 48]
        I49[Code block 49]
        I50[Code block 50]
        I51[Code block 51]
        I52[Code block 52]
        I53[Code block 53]
        I54[Code block 54]
        I55[Code block 55]
        I56[Code block 56]
        I57[Code block 57]
        I58[Code block 58]
        I59[Code block 59]
        I60[Code block 60]
        I61[Code block 61]
        I62[Code block 62]
        I63[Code block 63]
        I64[Code block 64]
        I65[Code block 65]
        I66[Code block 66]
        I67[Code block 67]
        I68[Code block 68]
        I69[Code block 69]
        I70[Code block 70]
        I71[Code block 71]
        I72[Code block 72]
        I73[Code block 73]
        I74[Code block 74]
        I75[Code block 75]
        I76[Code block 76]
        I77[Code block 77]
        I78[Code block 78]
        I79[Code block 79]
        I80[Code block 80]
        I81[Code block 81]
        I82[Code block 82]
        I83[Code block 83]
        I84[Code block 84]
        I85[Code block 85]
        I86[Code block 86]
        I87[Code block 87]
        I88[Code block 88]
        I89[Code block 89]
        I90[Code block 90]
        I91[Code block 91]
        I92[Code block 92]
        I93[Code block 93]
        I94[Code block 94]
        I95[Code block 95]
        I96[Code block 96]
        I97[Code block 97]
        I98[Code block 98]
        I99[Code block 99]
        I100[Code block 100]
    end
```

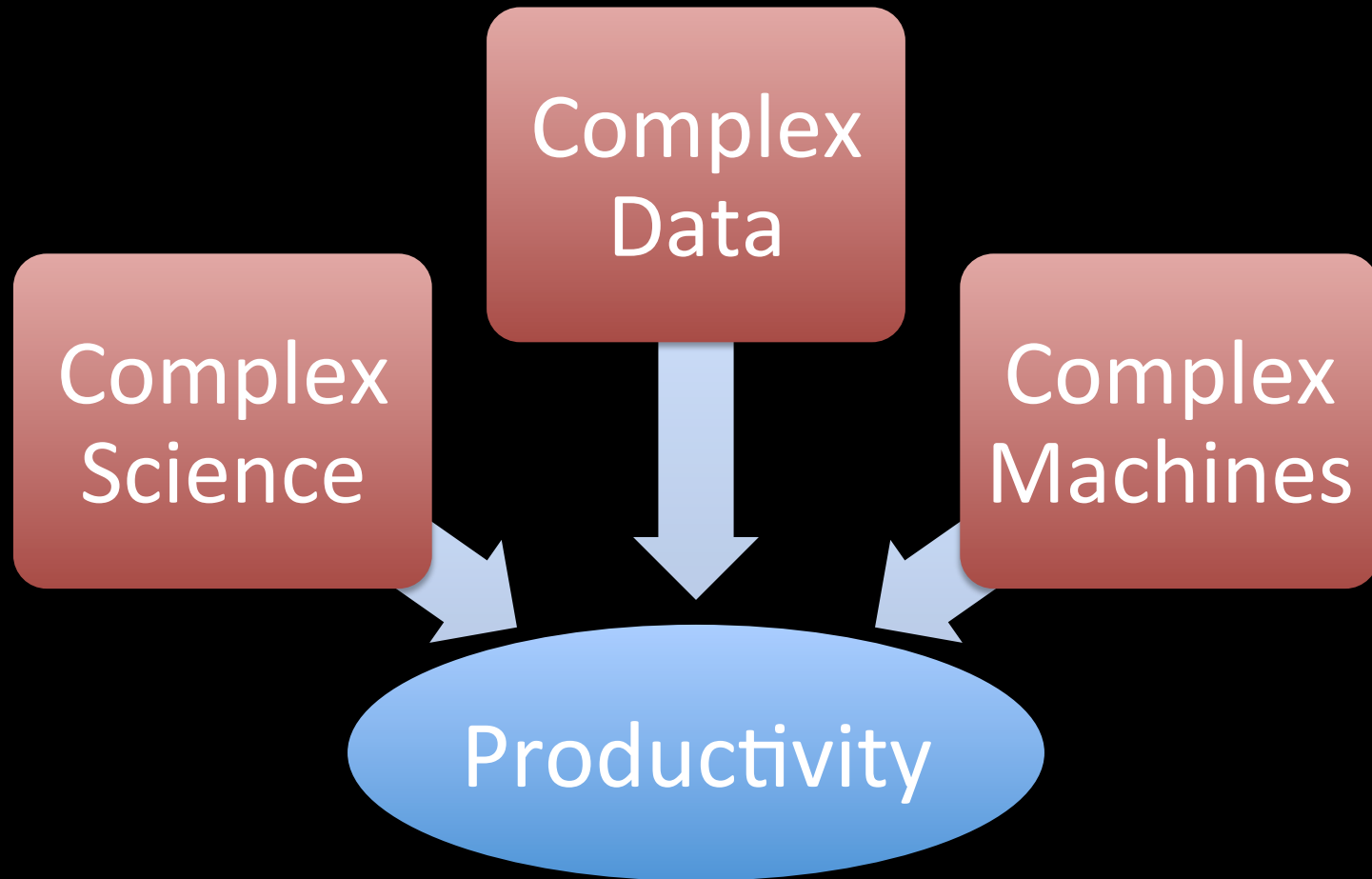
```
graph TD
    subgraph Rose_Compiler_Output [Rose compiler output]
        direction TB
        O1[Code block 1]
        O2[Code block 2]
        O3[Code block 3]
        O4[Code block 4]
        O5[Code block 5]
        O6[Code block 6]
        O7[Code block 7]
        O8[Code block 8]
        O9[Code block 9]
        O10[Code block 10]
        O11[Code block 11]
        O12[Code block 12]
        O13[Code block 13]
        O14[Code block 14]
        O15[Code block 15]
        O16[Code block 16]
        O17[Code block 17]
        O18[Code block 18]
        O19[Code block 19]
        O20[Code block 20]
        O21[Code block 21]
        O22[Code block 22]
        O23[Code block 23]
        O24[Code block 24]
        O25[Code block 25]
        O26[Code block 26]
        O27[Code block 27]
        O28[Code block 28]
        O29[Code block 29]
        O30[Code block 30]
        O31[Code block 31]
        O32[Code block 32]
        O33[Code block 33]
        O34[Code block 34]
        O35[Code block 35]
        O36[Code block 36]
        O37[Code block 37]
        O38[Code block 38]
        O39[Code block 39]
        O40[Code block 40]
        O41[Code block 41]
        O42[Code block 42]
        O43[Code block 43]
        O44[Code block 44]
        O45[Code block 45]
        O46[Code block 46]
        O47[Code block 47]
        O48[Code block 48]
        O49[Code block 49]
        O50[Code block 50]
        O51[Code block 51]
        O52[Code block 52]
        O53[Code block 53]
        O54[Code block 54]
        O55[Code block 55]
        O56[Code block 56]
        O57[Code block 57]
        O58[Code block 58]
        O59[Code block 59]
        O60[Code block 60]
        O61[Code block 61]
        O62[Code block 62]
        O63[Code block 63]
        O64[Code block 64]
        O65[Code block 65]
        O66[Code block 66]
        O67[Code block 67]
        O68[Code block 68]
        O69[Code block 69]
        O70[Code block 70]
        O71[Code block 71]
        O72[Code block 72]
        O73[Code block 73]
        O74[Code block 74]
        O75[Code block 75]
        O76[Code block 76]
        O77[Code block 77]
        O78[Code block 78]
        O79[Code block 79]
        O80[Code block 80]
        O81[Code block 81]
        O82[Code block 82]
        O83[Code block 83]
        O84[Code block 84]
        O85[Code block 85]
        O86[Code block 86]
        O87[Code block 87]
        O88[Code block 88]
        O89[Code block 89]
        O90[Code block 90]
        O91[Code block 91]
        O92[Code block 92]
        O93[Code block 93]
        O94[Code block 94]
        O95[Code block 95]
        O96[Code block 96]
        O97[Code block 97]
        O98[Code block 98]
        O99[Code block 99]
        O100[Code block 100]
    end
```

Rose compiler output

Halide DSL has 10+ developers
and 50+ users in industry



Productivity will be key to dealing with complexity



Conclusions

NSCI is **not**:

- limited to the high end
- about integrating enough chips to reach exaflop

It **is** about:

- The cost of computing
- Power requirements, size, and speed of computers
- Solving real problems that require more computing