eScience: Computational Science for the 21st Century

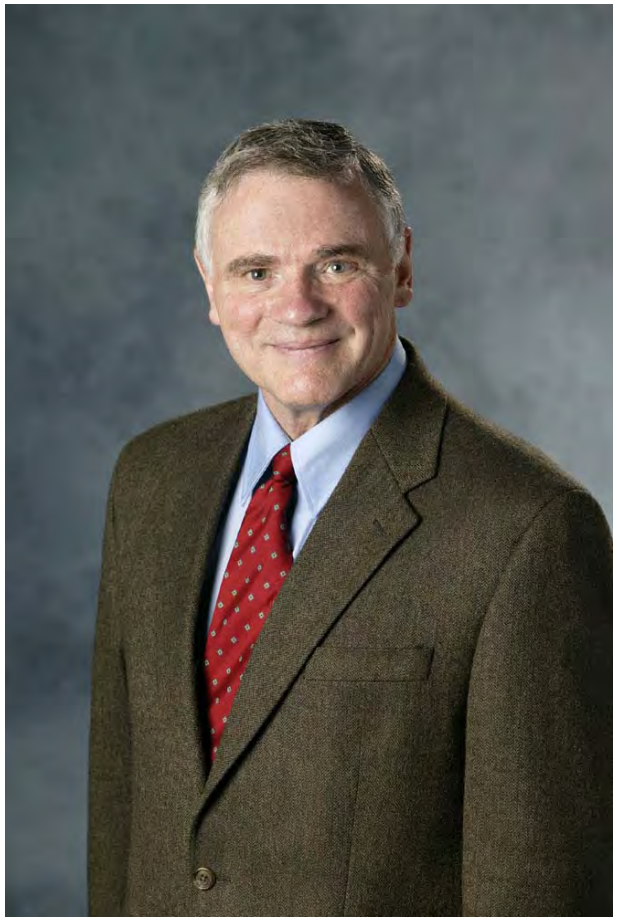
Ed Lazowska

Bill & Melinda Gates Chair in
Computer Science & Engineering
University of Washington

In Celebration of Lee Hood

October 2008

http://lazowska.cs.washington.edu/hood.pdf



DNA of Innovation Symposium & Gala

In Celebration of Lee Hood

Lee's visions

- The conjoining of biology and technology in a virtuous cycle
- The commercialization of the gene sequencer
- Biology viewed as an information science
- The launching of the Human Genome Project
- The responsibility of scientists to contribute to public understanding and education
- Systems biology and P4 medicine

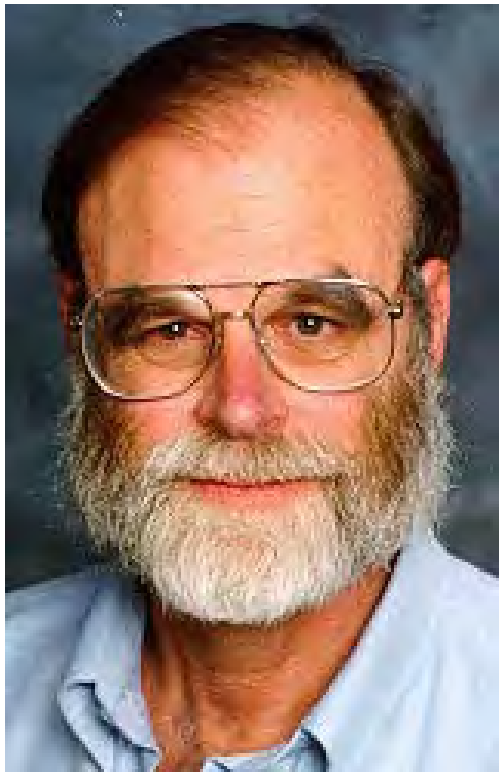
DNA *of* Innovation Symposium & Gala

In Celebration of Lee Hood

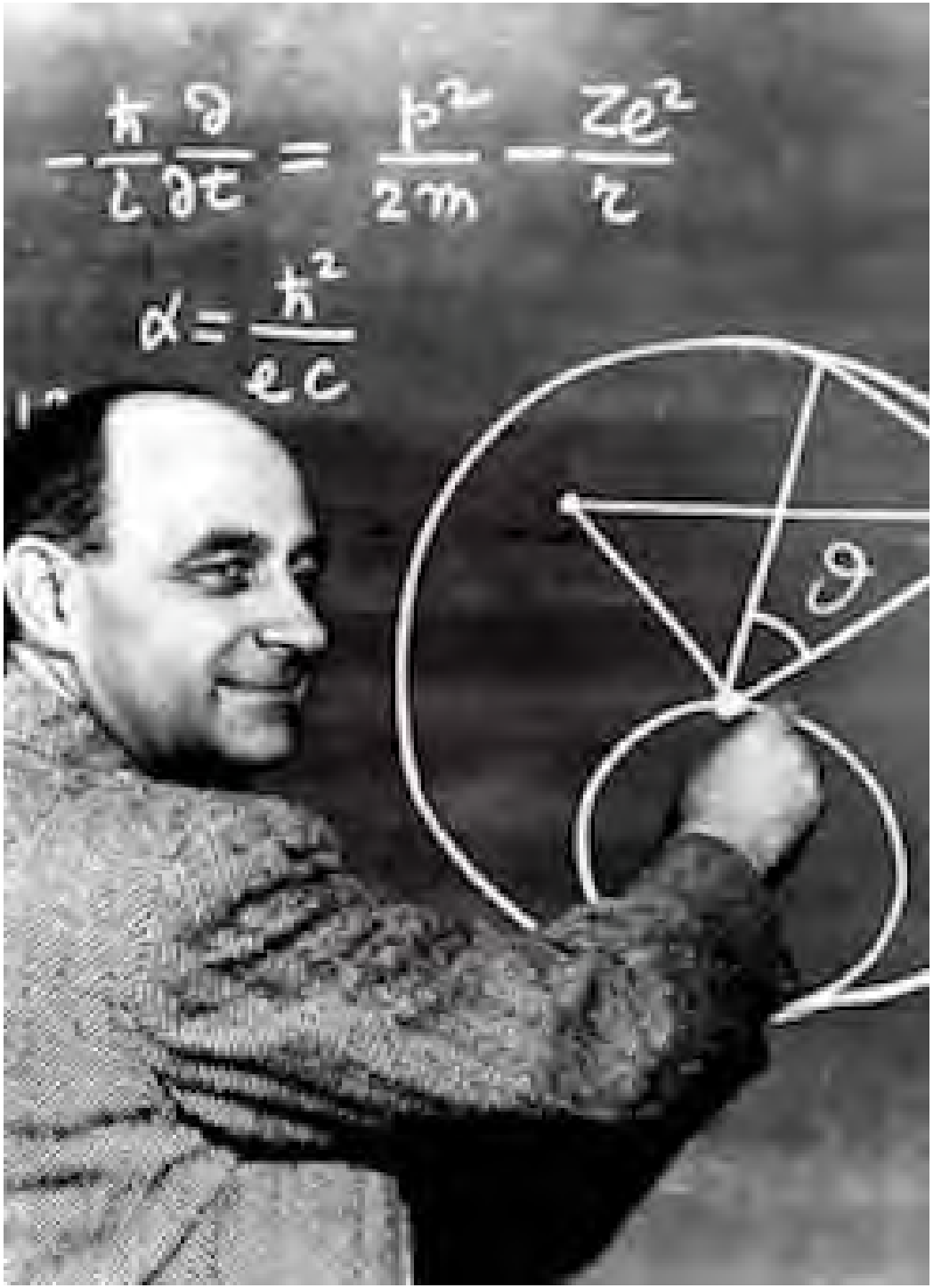
This afternoon

- eScience
- The technologies enabling it
- Grand challenges for engineering and computer science

eScience: Sensor-driven (data-driven) science and engineering



Transforming science (again!)



Theory

Experiment

Observation

Theory

Experiment

Observation

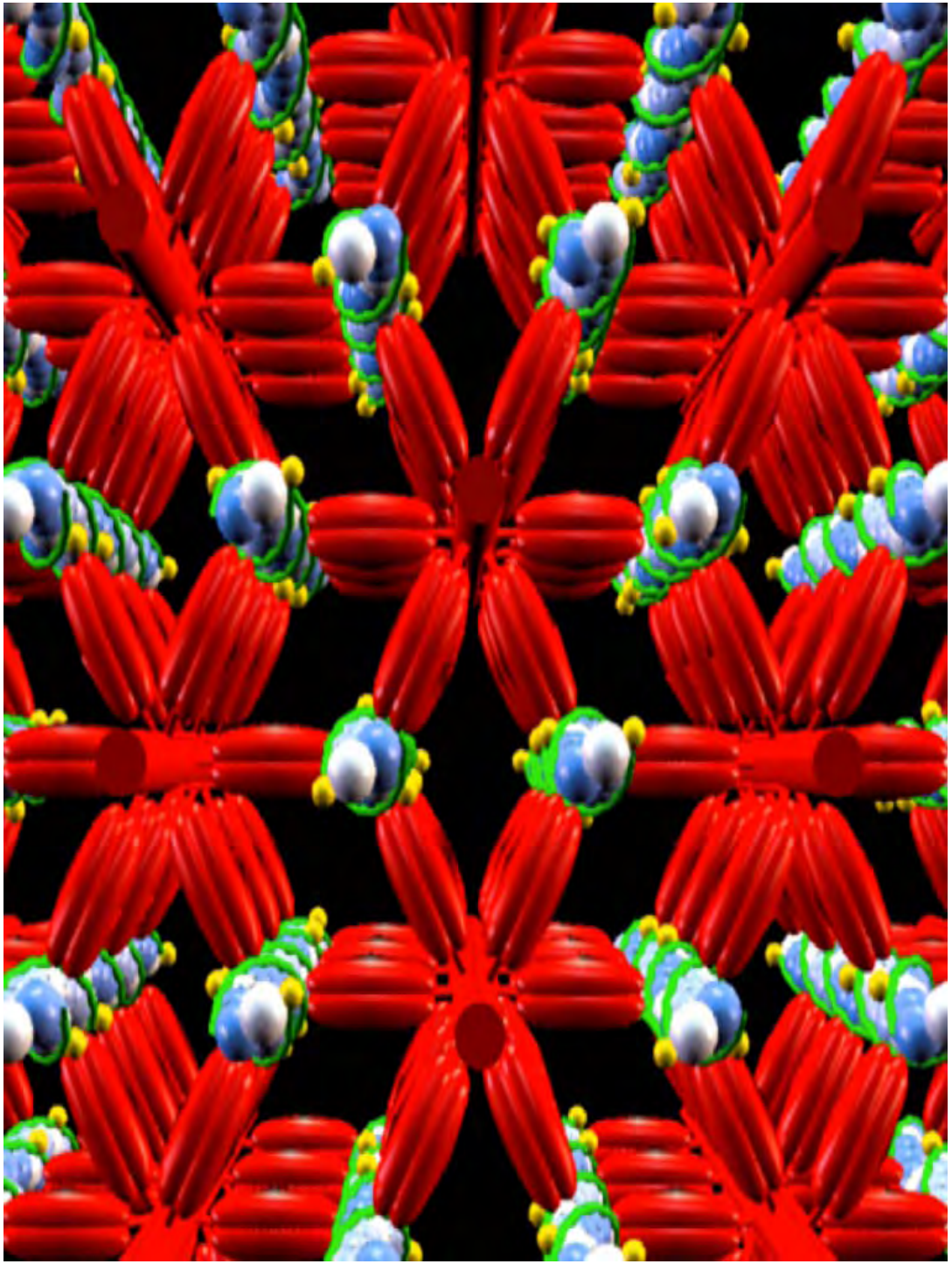

Theory
Experiment
Observation

Theory

Experiment

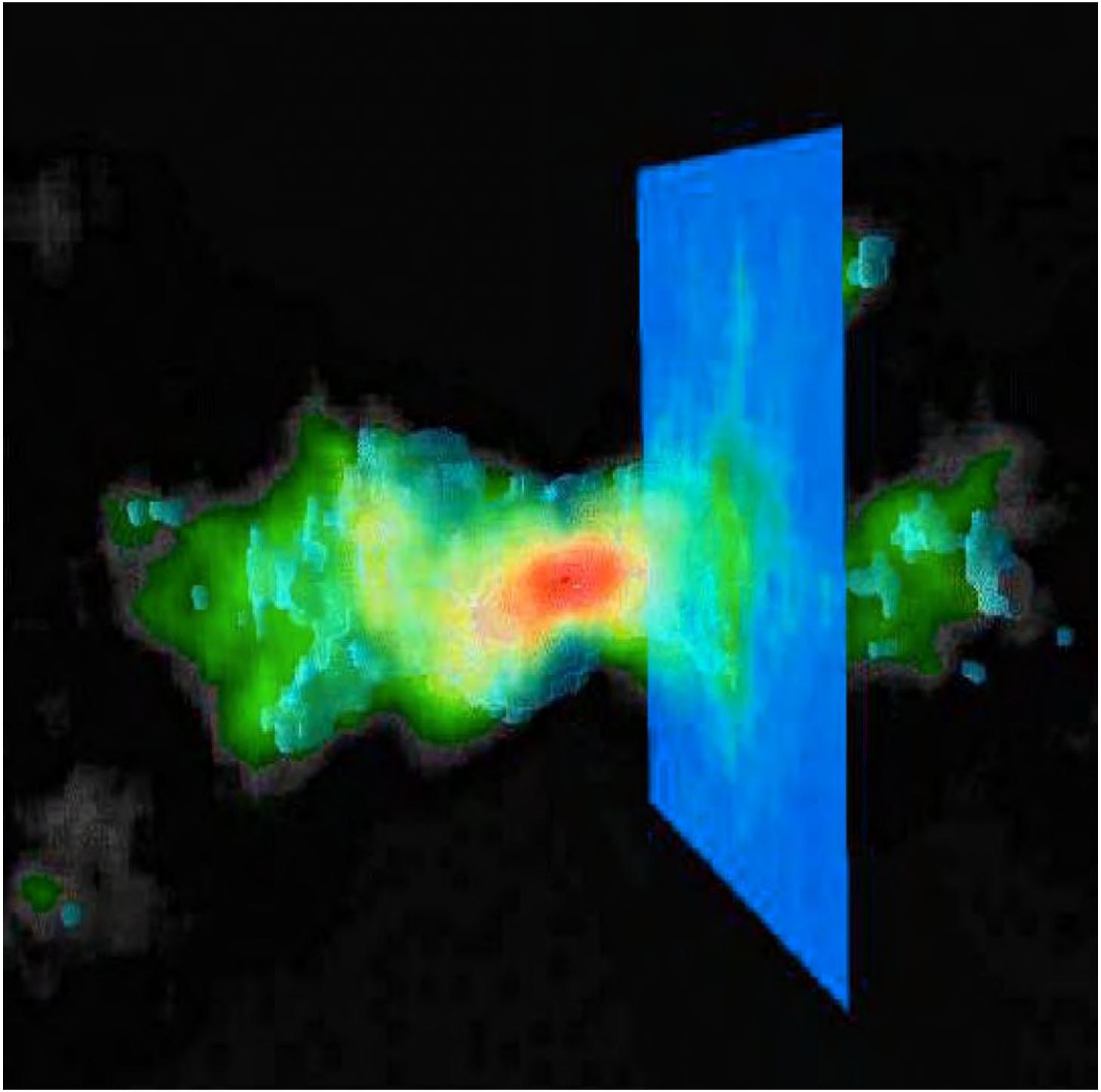
Observation

Computational Science



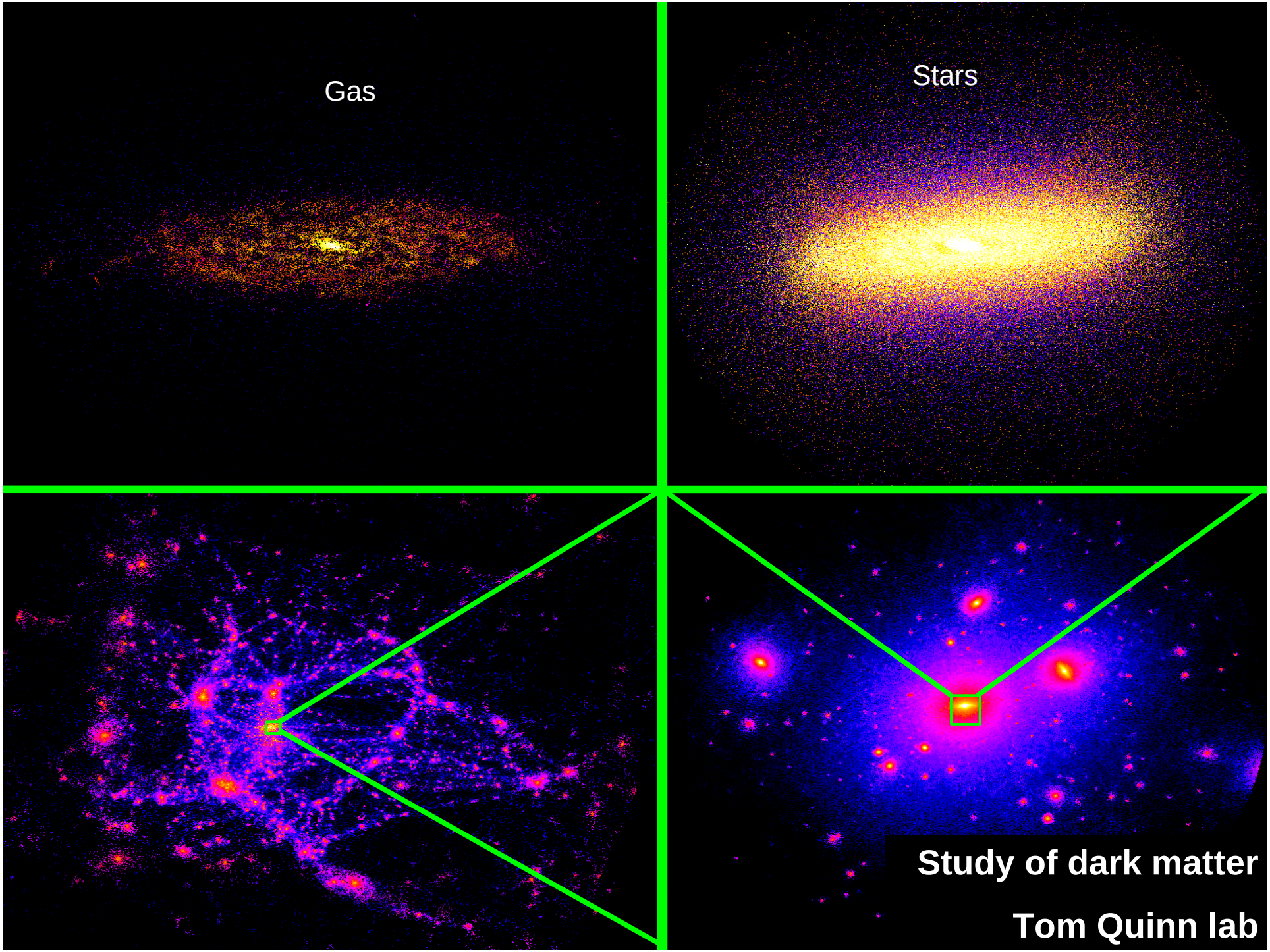
Protein interactions in striated muscles

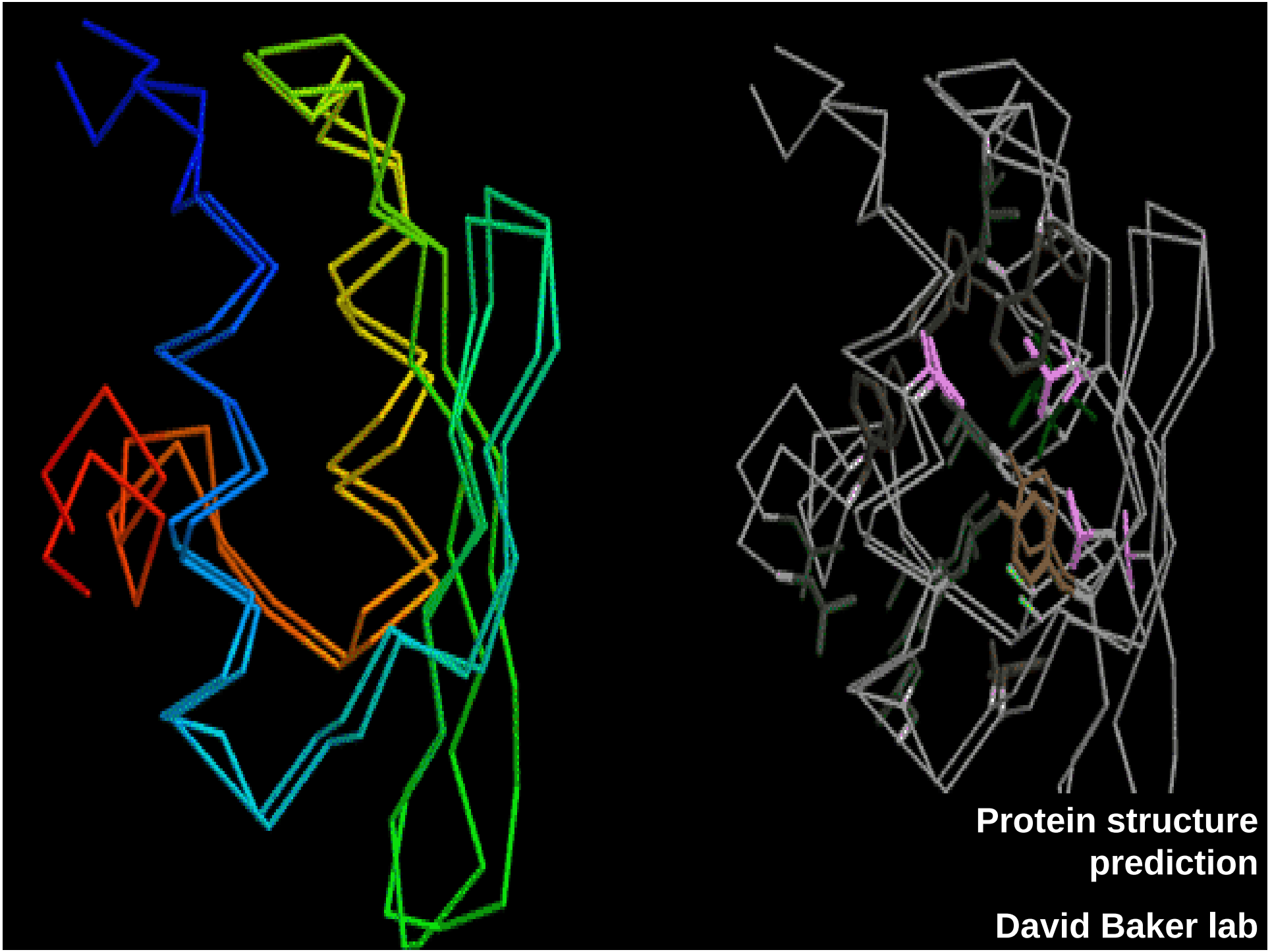
Tom Daniel lab



QCD to study interactions of nuclei

David Kaplan lab


Gas
Stars
Study of dark matter
Tom Quinn lab


Protein structure prediction
David Baker lab

Theory

Experiment

Observation

Computational Science

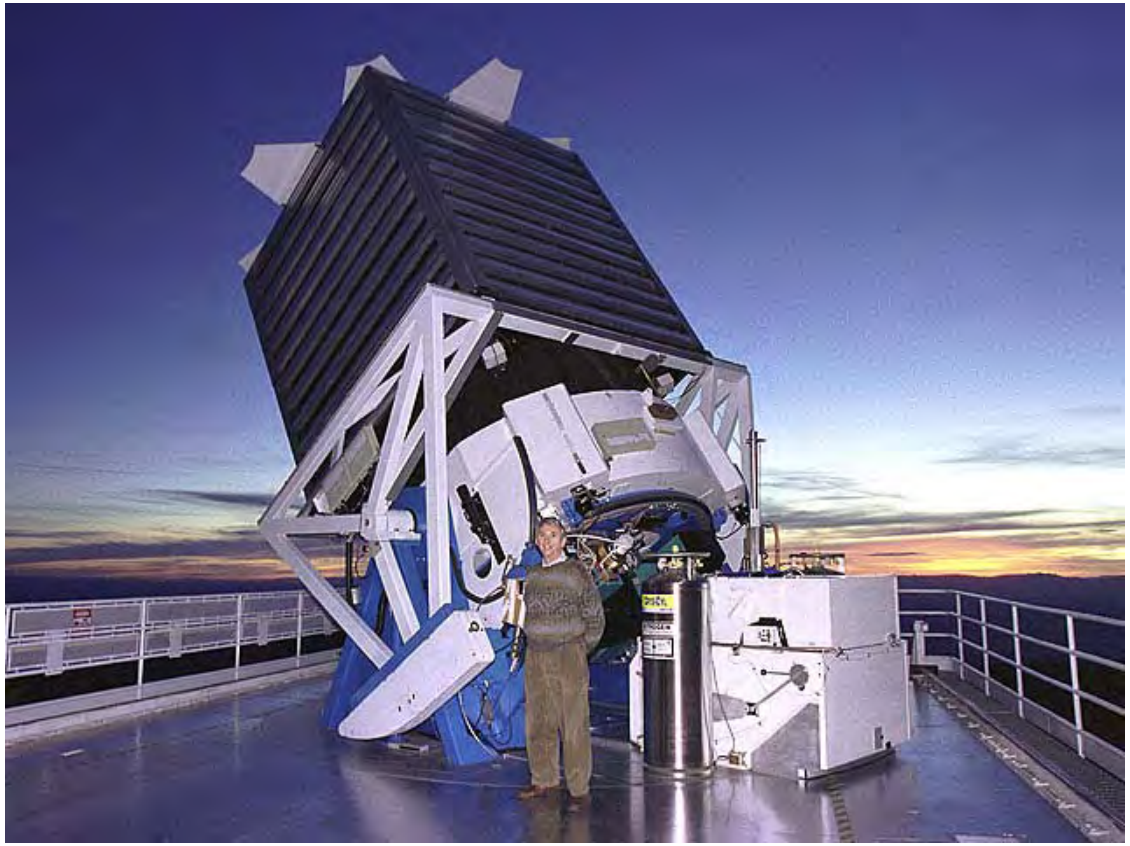
eScience

eScience is driven by *data*

- Massive volumes of data from sensors and networks of sensors

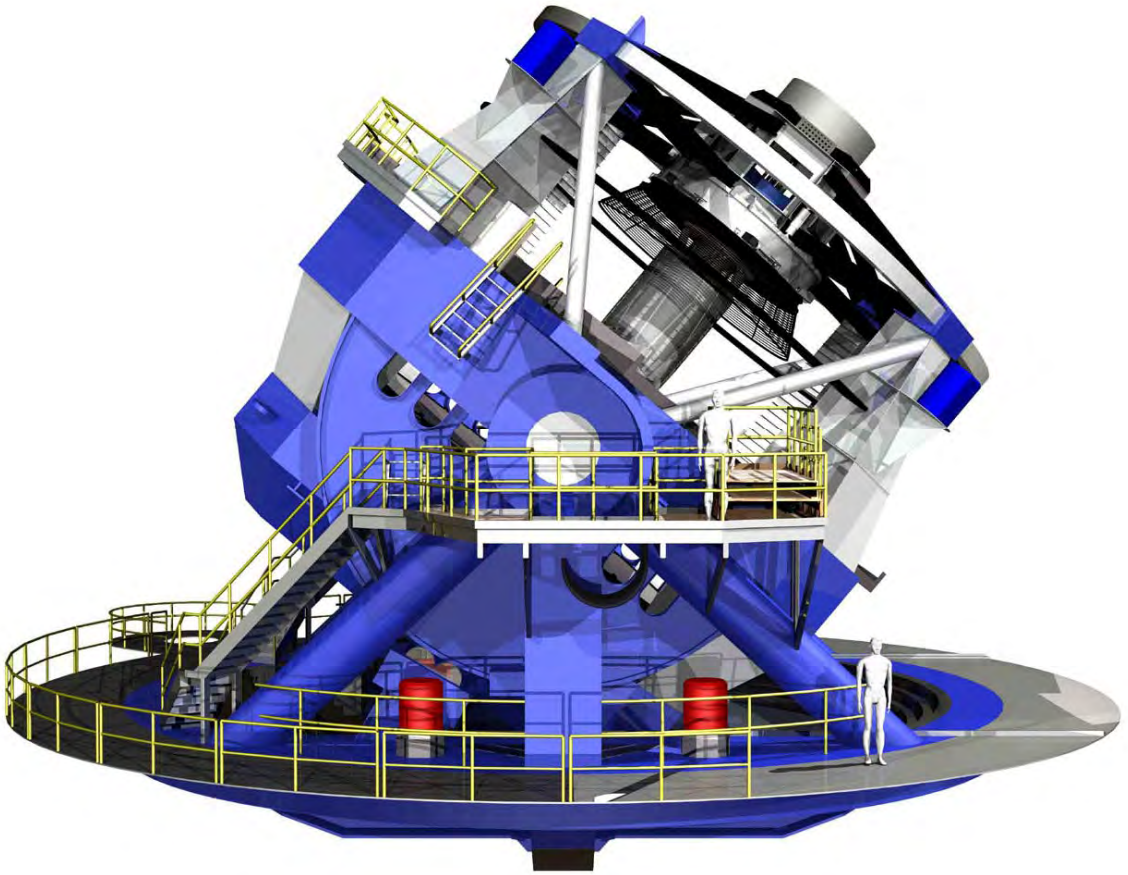
Apache Point telescope, SDSS

15TB of data (15,000,000,000,000 bytes)



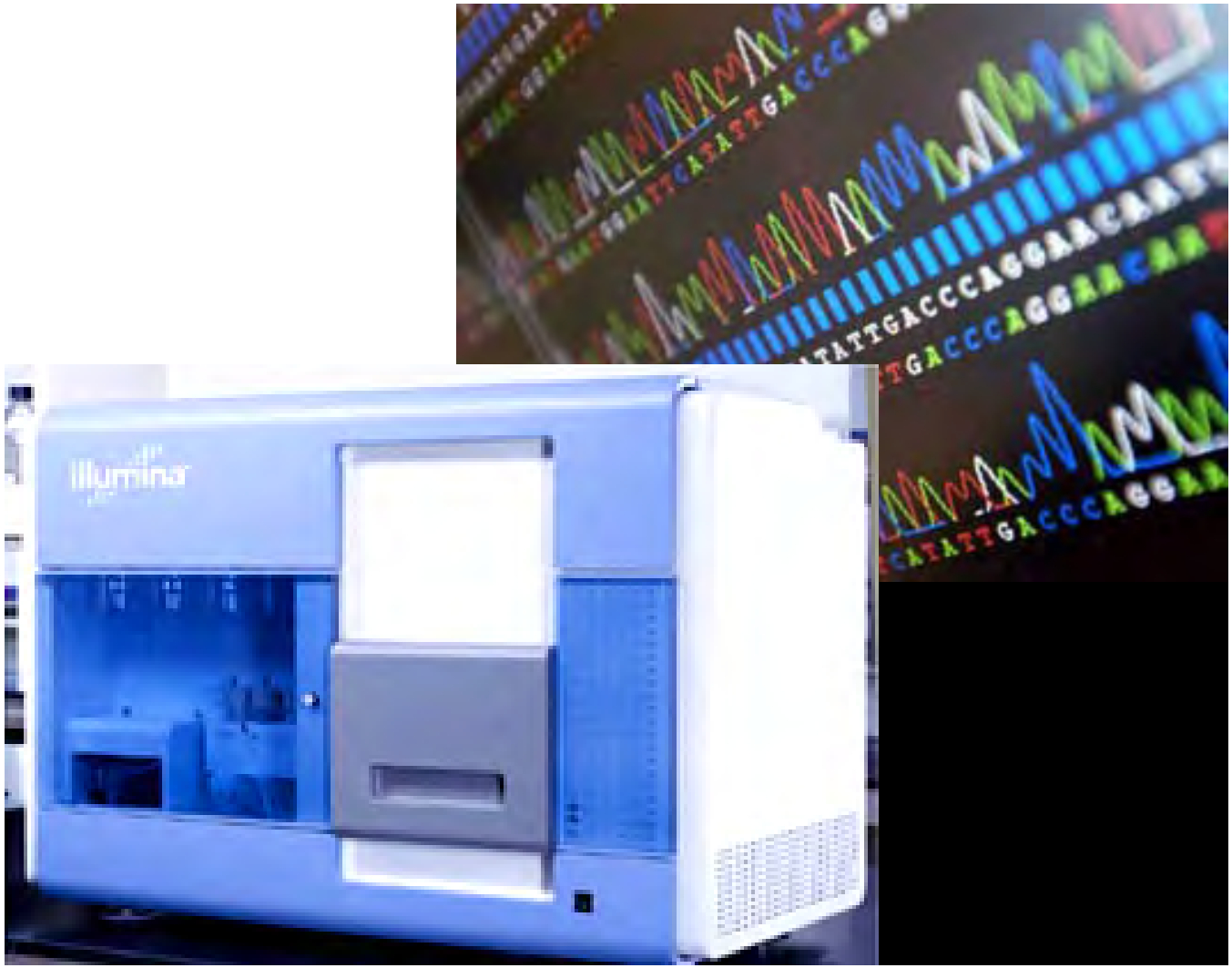
Large Synoptic Survey Telescope (LSST)

30TB/day, 60PB in its 10-year lifetime



Large Hadron Collider

700MB of data per second, 60TB/day, 20PB/year

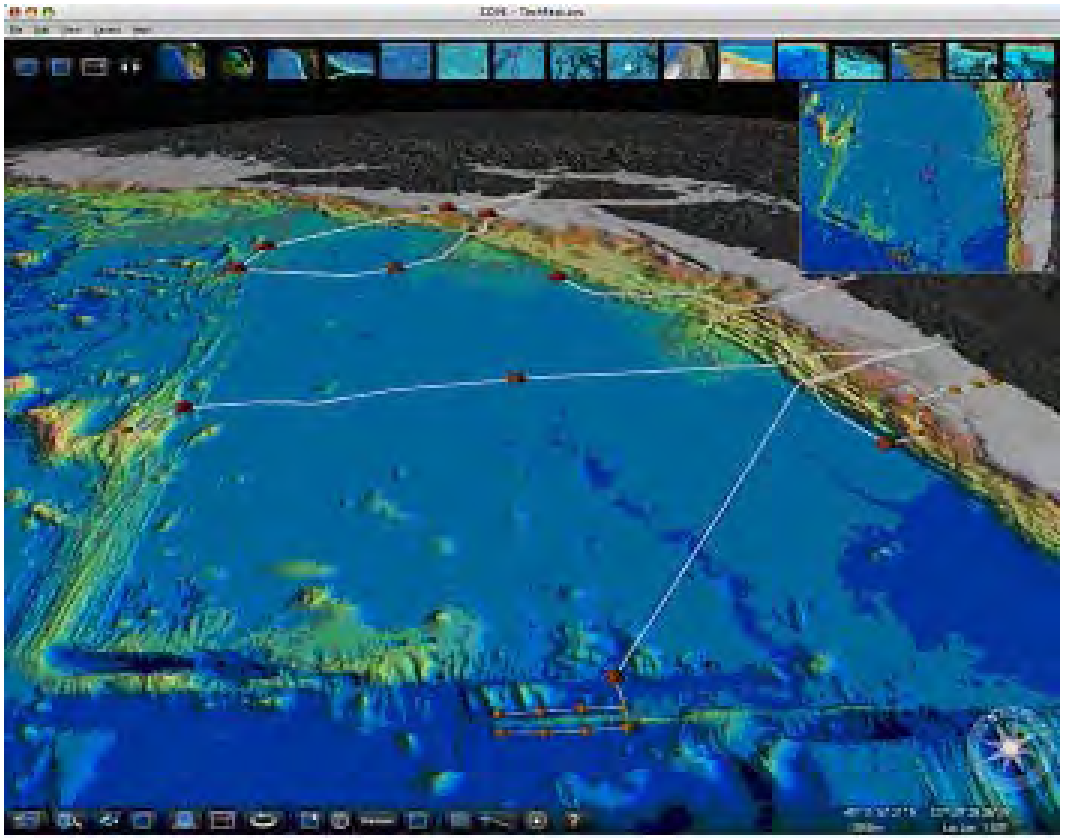


Illumina Genome Analyzer

~1TB/day

Regional Scale Nodes of the NSF Ocean Observatories Initiative

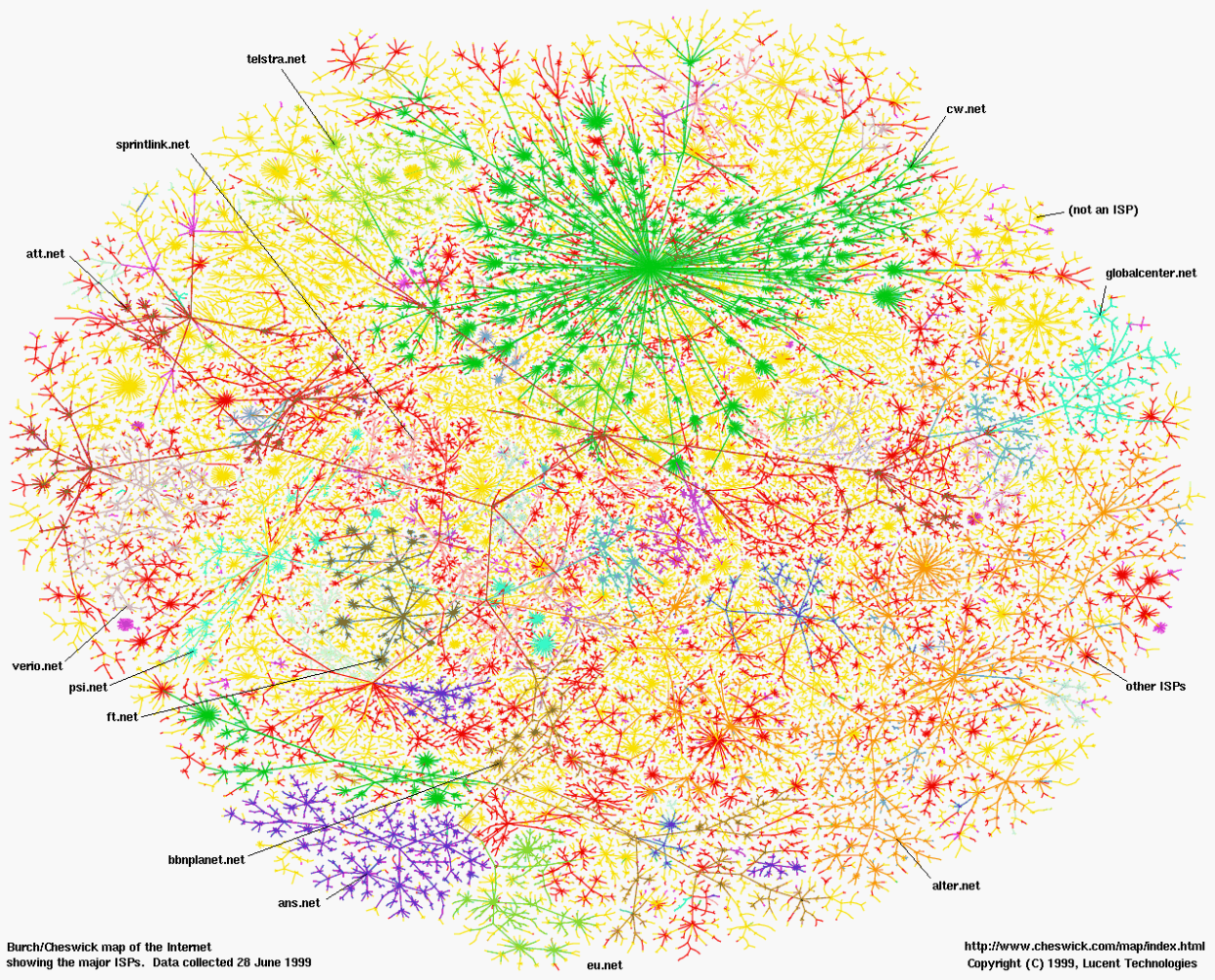
2000 km of fiber optic cable on the seafloor, connecting thousands of chemical, physical, and biological sensors



The Web

20+ billion web pages x 20KB = 400+TB

One computer can read 30-35 MB/sec from disk => 4 months just to read the web



Point-of-sale terminals

eScience is about the *analysis* of data

- The automated or semi-automated extraction of knowledge from massive volumes of data
  - There's simply too much of it to look at

The technologies of eScience

- Sensors and sensor networks
- Databases
- Data mining
- Machine learning
- Data visualization

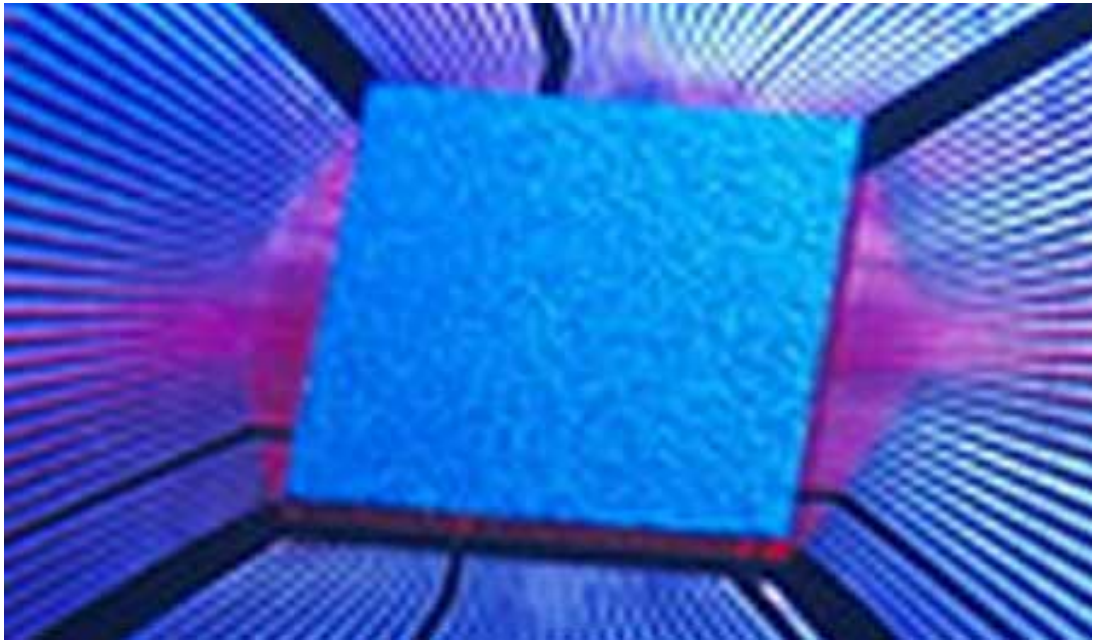
eScience will be pervasive

- Computational science was a niche
  - As an institution (e.g., a university), you didn't need to excel in order to be competitive
- eScience capabilities must be broadly available in any organization
  - If not, the organization will simply cease to be competitive

More about the enablement of eScience

- Ten quintillion: $10*10^{18}$
  - The number of grains of rice harvested in 2004

- Ten quintillion: $10*10^{18}$
  - The number of grains of rice harvested in 2004
  - The number of transistors fabricated in 2004



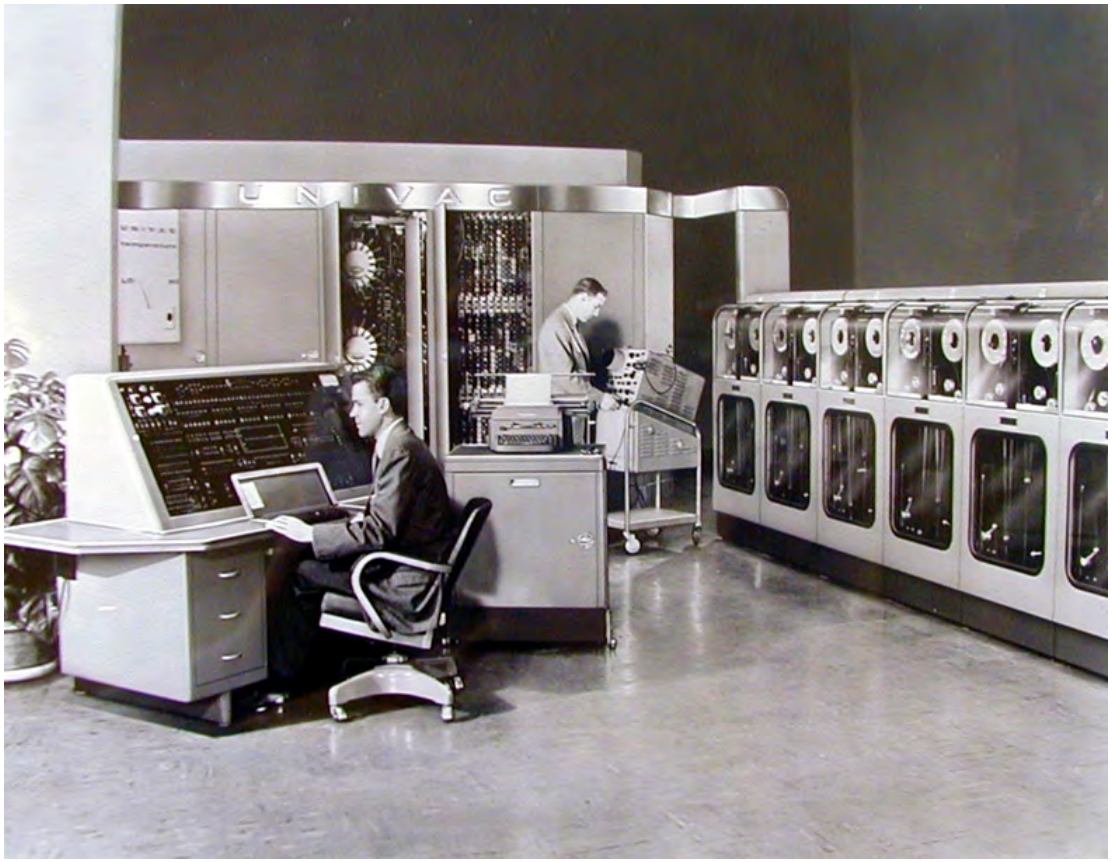
The transistor

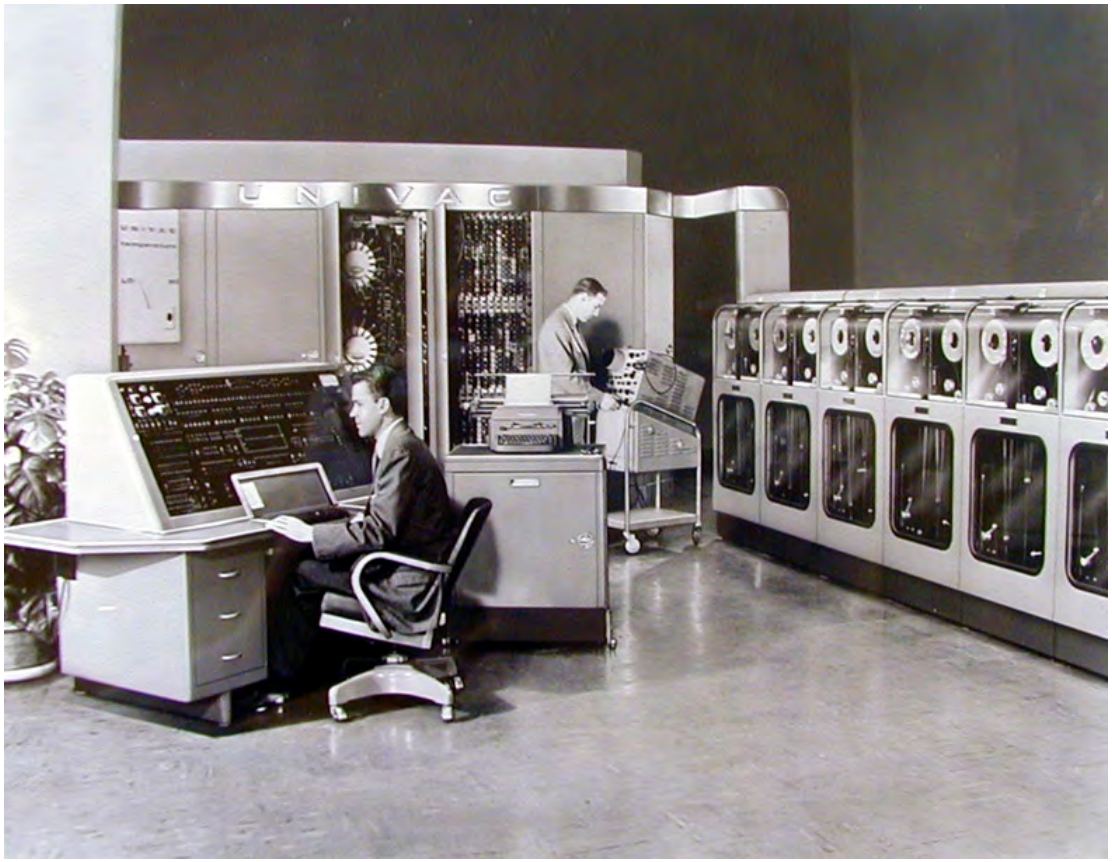
- William Shockley, Walter Brattain and John Bardeen, Bell Labs, 1947

- The integrated circuit
  - Jack Kilby, Texas Instruments, and Bob Noyce, Fairchild Semiconductor Corporation, 1958

Moore's Law

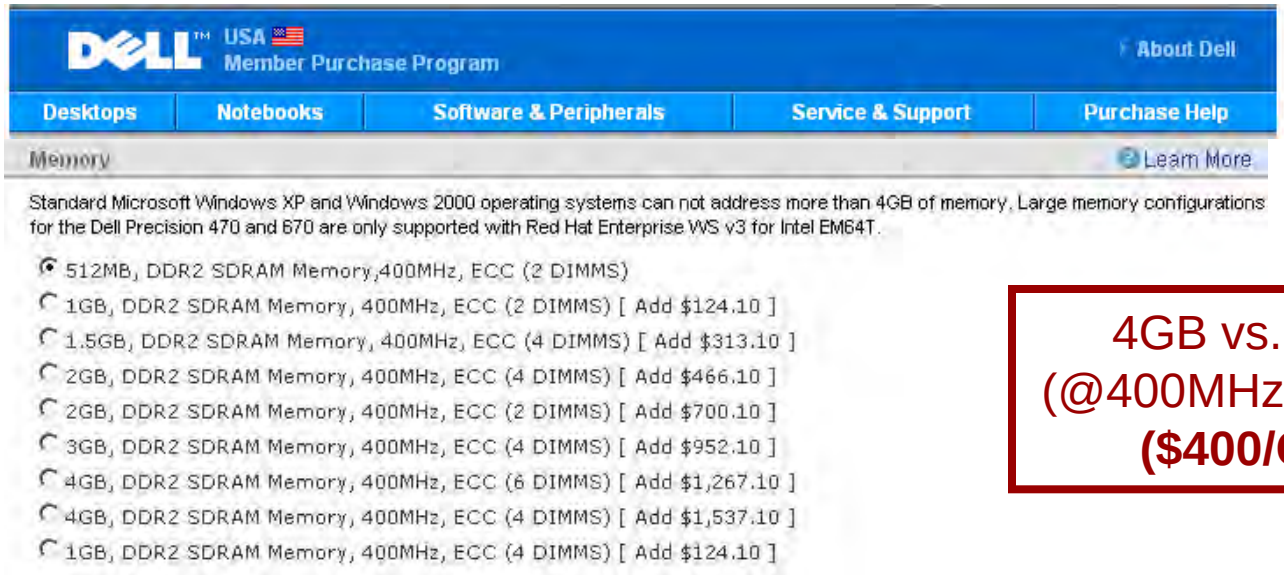
- Gordon Moore, 1965


UNIVAC


UNIVAC

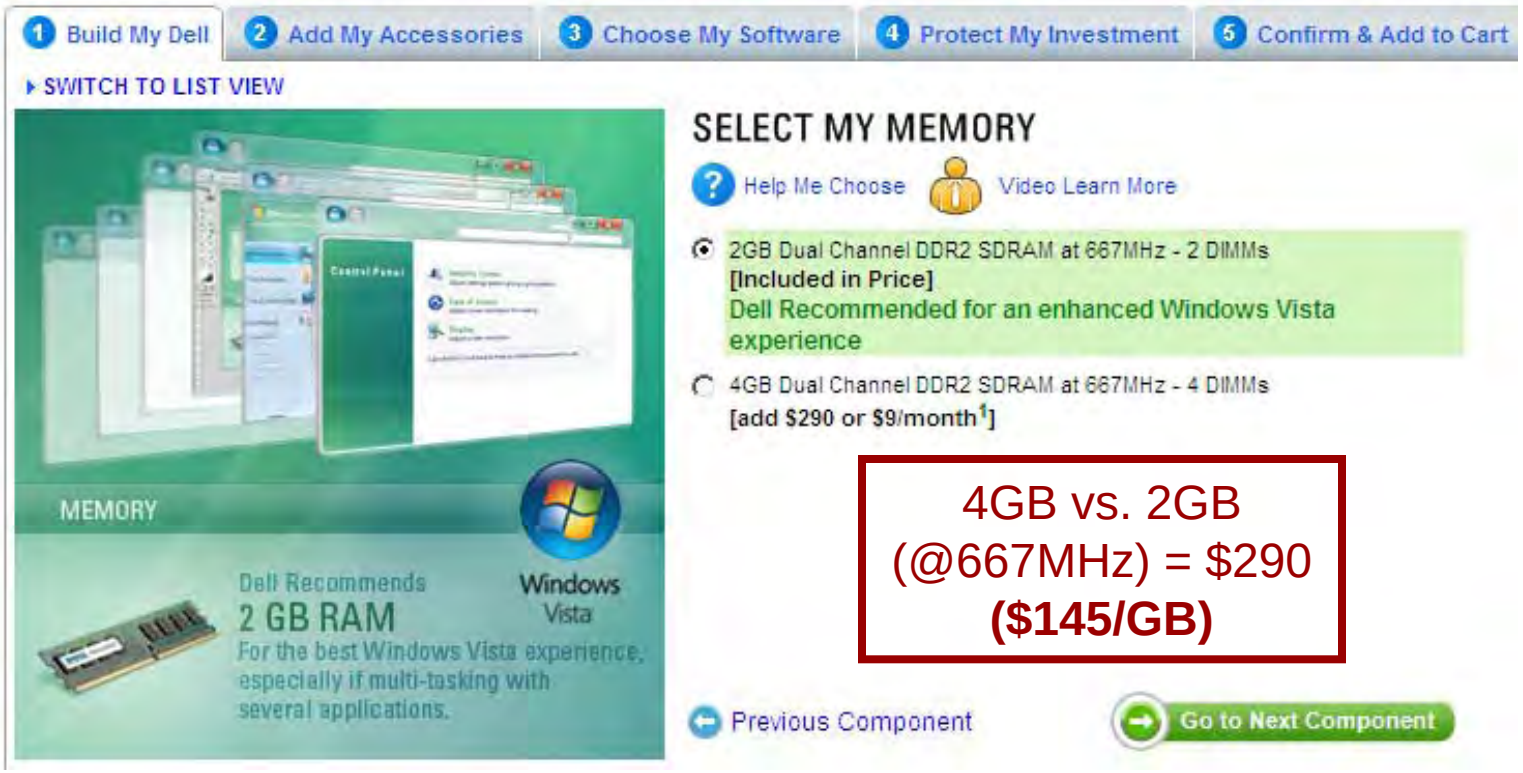
- Processing power, historically
  - 1980: 1 MHz Apple II+, $2,000
    - 1980 also 1 MIPS VAX-11/780, $120,000
  - 2006: 2.4 GHz Pentium D, $800
    - A factor of 6000
- Processing power, recently
  - Additional transistors => more cores of the same speed, rather than higher speed
  - 2008: Intel Core 2 Quad-Core 2.4 GHz, $800

- Primary memory – same story, same reason (but no multicore fiasco)
  - 1972: 1MB, $1,000,000
  - 1982: 1MB, $60,000
  - 2005: $400/GB (1MB, $0.40)



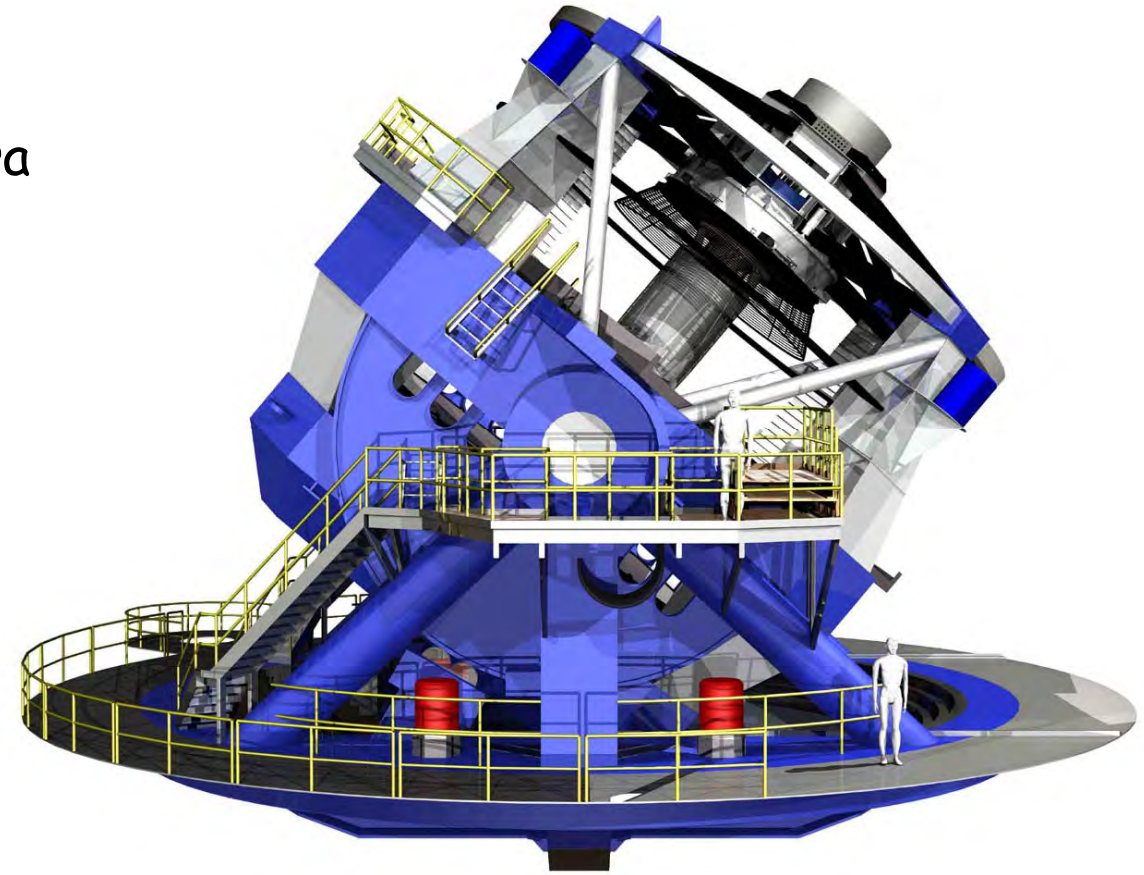
4GB vs. 2GB (@400MHz) = $800 **($400/GB)**

- 2007: $145/GB (1MB, $0.15)



- 2008: $49/GB (1MB, $0.05)

- Moore's Law drives sensors as well as processing and memory
  - LSST will have a 3.2 Gigapixel camera



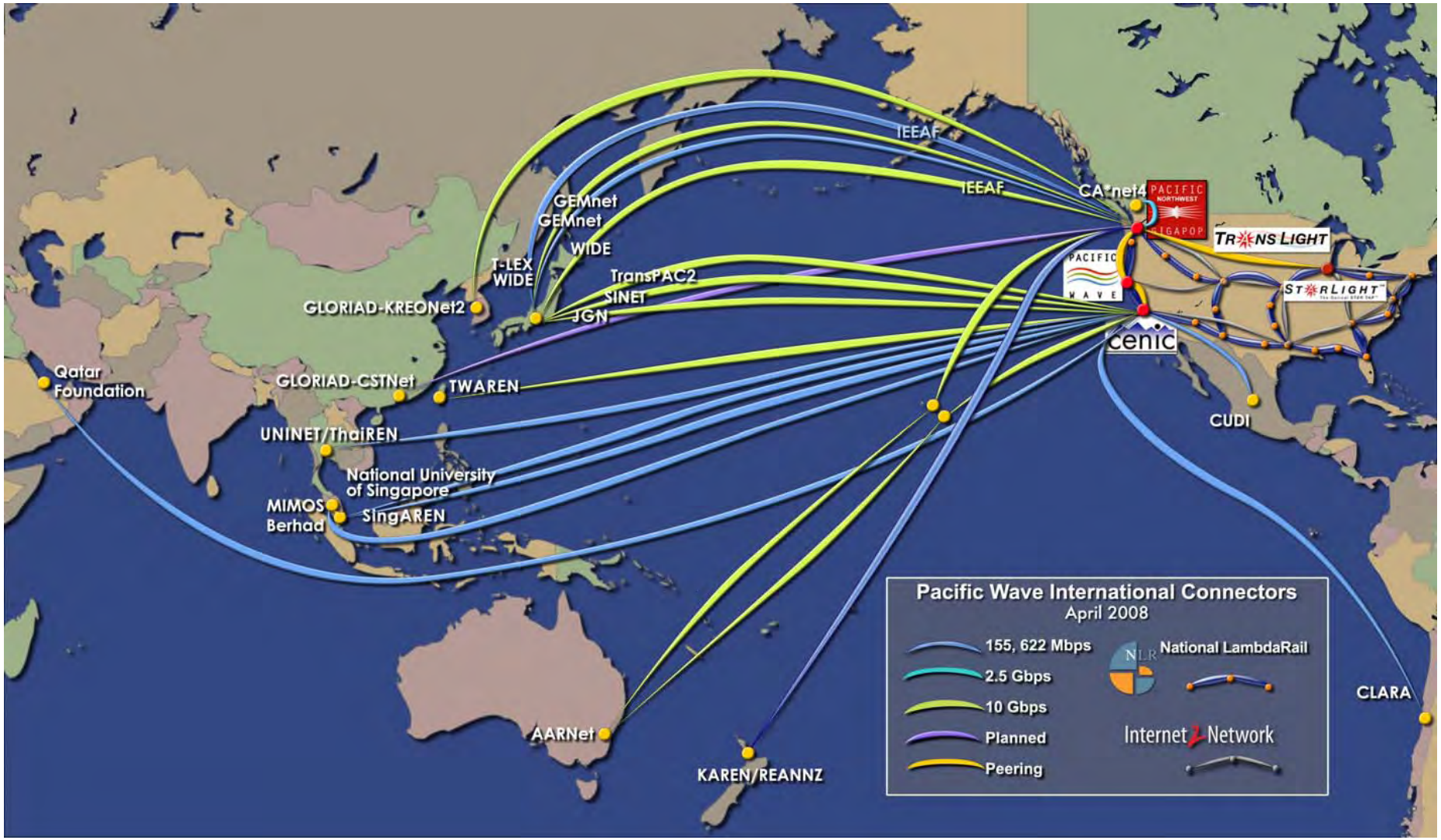
- Disk capacity, 1975-1989
  - doubled every 3+ years
  - 25% improvement each year
  - factor of 10 every decade
  - Still exponential, but far less rapid than processor performance
- Disk capacity since 1990
  - doubling every 12 months
  - 100% improvement each year
  - factor of 1000 every decade
  - 10x as fast as processor performance

- Only a few years ago, we purchased disks by the megabyte (and it hurt!)
- Current cost of 1 GB (a billion bytes) from Dell
  - 2005: $1.00
  - 2006: $0.50
  - 2008: $0.25
- Purchase increment
  - 2005: 40GB
  - 2006: 80GB
  - 2008: 250GB

- Optical bandwidth today
  - Doubling every 9 months
  - 150% improvement each year
  - Factor of 10,000 every decade
  - 10x as fast as disk capacity
  - 100x as fast as processor performance

A connected region – then

A connected region – now

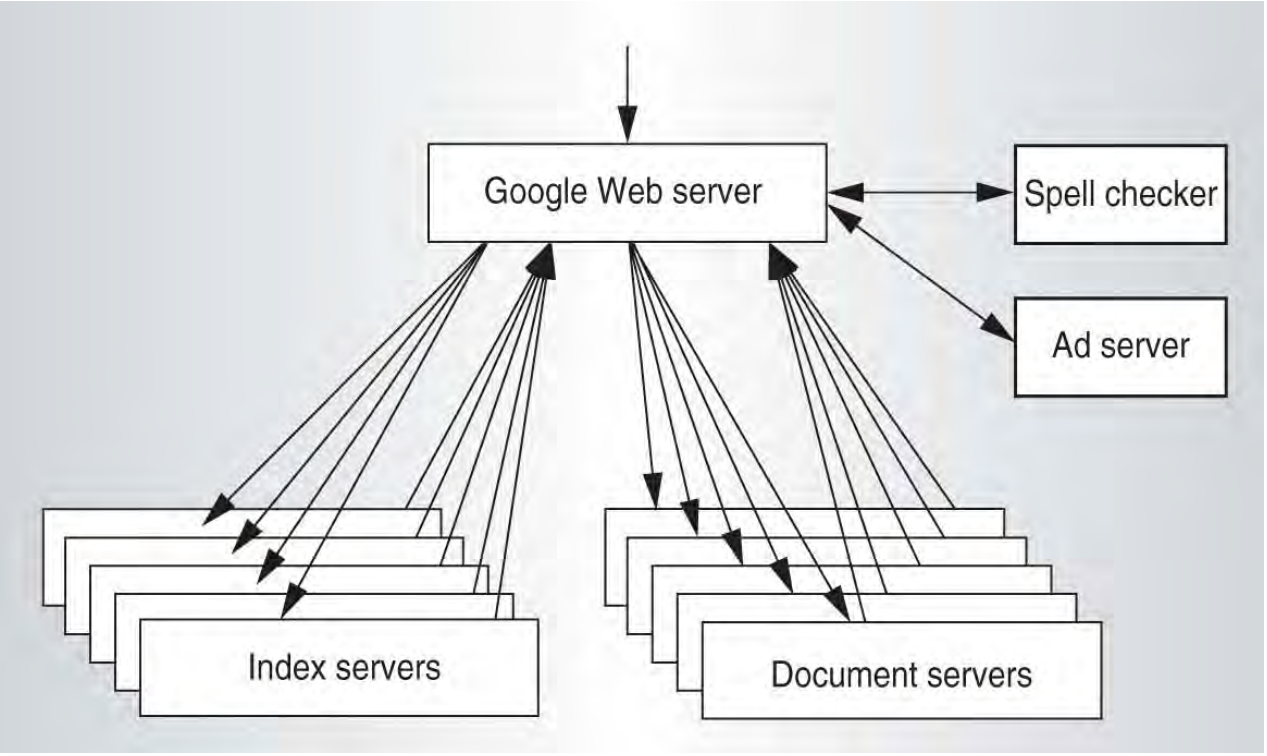


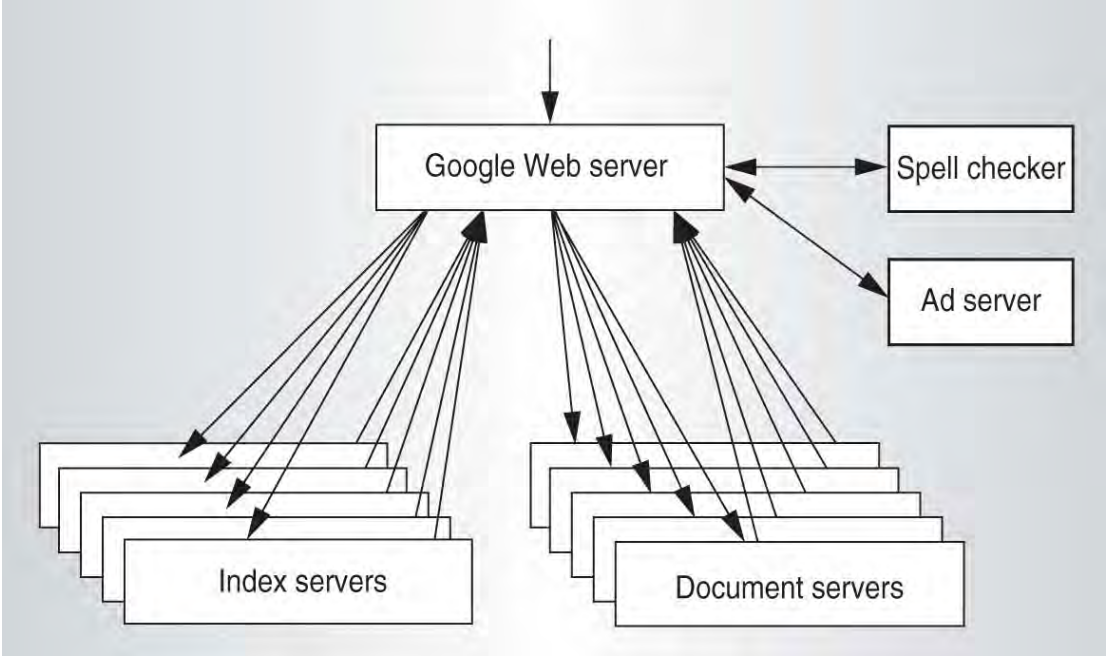
But eScience is equally enabled by *software* for *scalability* and for *discovery*

- It's likely that Google has several million machines
  - But let's be conservative – 1,000,000 machines
  - A rack holds 176 CPUs (88 1U dual-processor boards), so that's about 6,000 racks
  - A rack requires about 50 square feet (given datacenter cooling capabilities), so that's about 300,000 square feet of machine room space (more than 6 football fields of real estate – although of course Google divides its machines among dozens of datacenters all over the world)
  - A rack requires about 10kw to power, and about the same to cool, so that's about 120,000 kw of power, or nearly 100,000,000 kwh per month ($10 million at $0.10/kwh)
    - Equivalent to about 20% of Seattle City Light's generating capacity

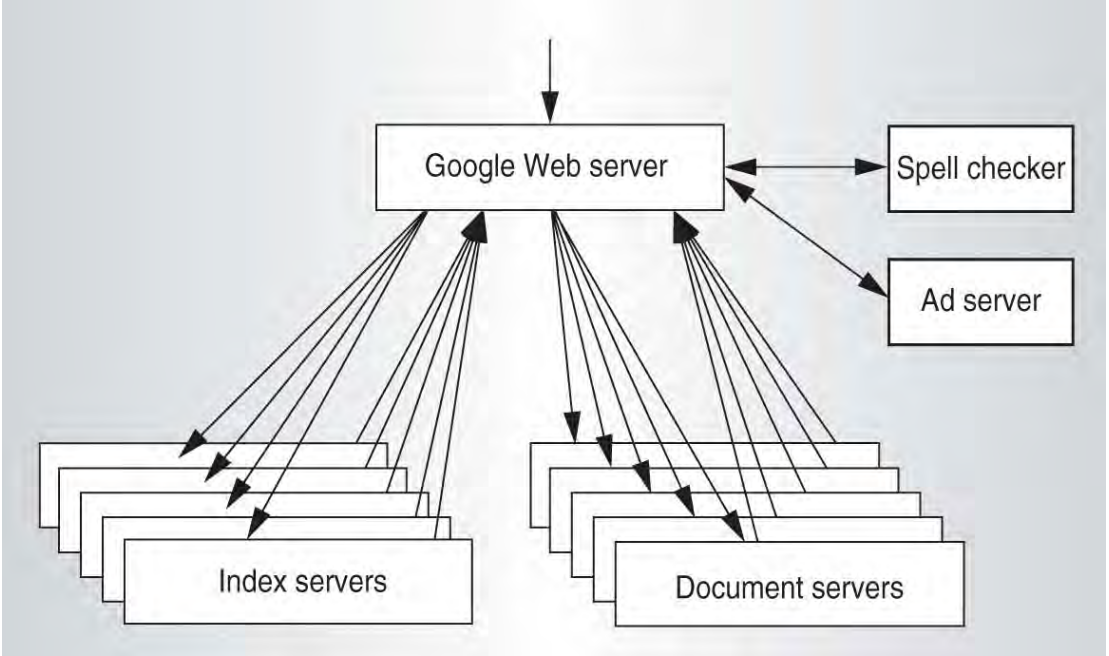
- Many hundreds of machines are involved in a single Google search request (remember, the web is 400+TB)
  - There are multiple clusters (of thousands of computers each) all over the world
  - DNS routes your search to a nearby cluster

- A cluster consists of Google Web Servers, Index Servers, Doc Servers, and various other servers (ads, spell checking, etc.)
- These are cheap standalone computers, rack-mounted, connected by commodity networking gear

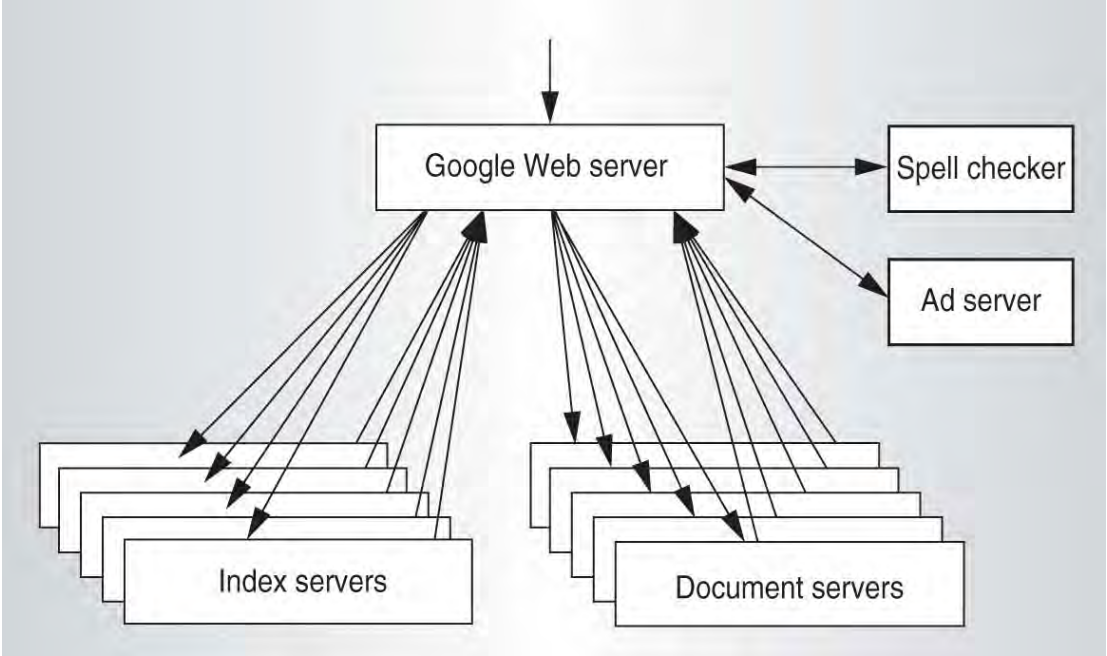




- Within the cluster, load-balancing routes your search to a lightly-loaded Google Web Server (GWS), which will coordinate the search and response
- The index is partitioned into "shards." Each shard indexes a subset of the docs (web pages). Each shard is replicated, and can be searched by multiple computers – "index servers"
- The GWS routes your search to one index server associated with each shard, through another load-balancer
- When the dust has settled, the result is an ID for every doc satisfying your search, rank-ordered by relevance



- The docs, too, are partitioned into "shards" – the partitioning is a hash on the doc ID. Each shard contains the full text of a subset of the docs. Each shard can be searched by multiple computers – "doc servers"
- The GWS sends appropriate doc IDs to one doc server associated with each relevant shard
- When the dust has settled, the result is a URL, a title, and a summary for every relevant doc



- Meanwhile, the ad server has done its thing, the spell checker has done its thing, etc.
- The GWS builds an HTTP response to your search and ships it off

- Many hundreds of computers have enabled you to search 400+TB of web in ~100 ms.

- Enormous volumes of data
- Extreme parallelism
- The cheapest imaginable components
  - Failures occur all the time
  - You couldn't afford to prevent this in hardware
- Software makes it
  - Fault-Tolerant
  - Highly Available
  - Recoverable
  - Consistent
  - Scalable
  - Predictable
  - Secure

How on earth would you enable mere mortals write hairy applications such as this?

- Recognize that many Google applications have the same structure
  - Apply a "map" operation to each logical record in order to compute a set of intermediate key/value pairs
  - Apply a "reduce" operation to all the values that share the same key in order to combine the derived data appropriately
- Example: Count the number of occurrences of each word in a large collection of documents
  - Map: Emit <word, 1> each time you encounter a word
  - Reduce: Sum the values for each word

- Build a runtime library that handles all the details, accepting a couple of customization functions from the user – a Map function and a Reduce function
- That's what MapReduce is
  - Supported by the Google File System and the Chubby lock manager
  - Augmented by the BigTable not-quite-a-database system
- Does your application run in this environment?
  - If not, figure out how to make it do so!

NATIONAL ACADEMY OF ENGINEERING
OF THE NATIONAL ACADEMIES

GRAND CHALLENGES FOR ENGINEERING


Make solar energy economical


Provide energy from fusion


Develop carbon sequestration methods


Manage the nitrogen cycle


Provide access to clean water


Restore and improve urban infrastructure


Advance health informatics


Engineer better medicines


Reverse-engineer the brain


Prevent nuclear terror


Secure cyberspace


Enhance virtual reality


Advance personalized learning


Engineer the tools of scientific discovery

NATIONAL ACADEMY OF ENGINEERING
OF THE NATIONAL ACADEMIES

CHALLENGES | IDEAS | NEXT STEPS | COMMITTEE

GRAND CHALLENGES FOR ENGINEERING


Make solar energy economical


Provide energy from fusion


Develop carbon sequestration methods


Manage the nitrogen cycle


Provide access to clean water


Restore and improve urban infrastructure


Advance health informatics


Engineer better medicines


Reverse-engineer the brain


Prevent nuclear terror


Secure cyberspace


Enhance virtual reality


Advance personalized learning


Engineer the tools of scientific discovery

NATIONAL ACADEMY OF ENGINEERING
OF THE NATIONAL ACADEMIES

CHALLENGES | IDEAS | NEXT STEPS | COMMITTEE

GRAND CHALLENGES FOR ENGINEERING


Make solar energy economical


Provide energy from fusion


Develop carbon sequestration methods


Manage the nitrogen cycle


Provide access to clean water


Restore and improve urban infrastructure


Advance health informatics


Engineer better medicines


Reverse-engineer the brain


Prevent nuclear terror


Secure cyberspace


Enhance virtual reality


Advance personalized learning


Engineer the tools of scientific discovery

NATIONAL ACADEMY OF ENGINEERING
OF THE NATIONAL ACADEMIES

CHALLENGES | IDEAS | NEXT STEPS | COMMITTEE

GRAND CHALLENGES FOR ENGINEERING


Make solar energy economical


Provide energy from fusion


Develop carbon sequestration methods


Manage the nitrogen cycle


Provide access to clean water


Restore and improve urban infrastructure


Advance health informatics


Engineer better medicines


Reverse-engineer the brain


Prevent nuclear terror


Secure cyberspace


Enhance virtual reality


Advance personalized learning


Engineer the tools of scientific discovery

Predominant CS component

Significant CS component

The bottom line …

- The future really couldn't be brighter
  - Well, ignoring Iraq, the economy, the election, and the failure of our education system