Parallel Programming

Lecture 01: Introduction to Parallel Computing

谭光明　研究员、博士生导师
高性能计算机研究中心，中科院计算技术研究所
计算机体系结构国家重点实验室

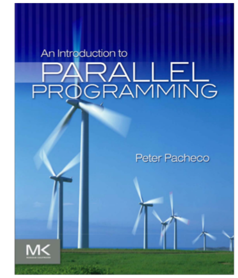
http://www.ncic.ac.cn/~tgm

Teacher's Team

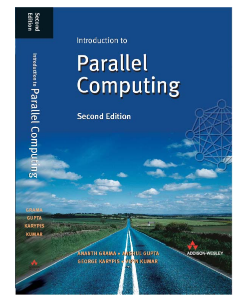
- 谭光明，Professor，tgm@ict.ac.cn
- 王银山，Assistant Professor，wangyinshan@ncic.ac.cn
- 严煜瑾，Ph.D Student，yanyujin@ncic.ac.cn

References

- **Peter S. Pacheco**. An Introduction to Parallel Programming.



- **Grama, A., Gupta, A., Kumar V.** (2003, 2nd edn.). Introduction to Parallel Computing



- **David E. Culler, Jaswinder Pal Singh**. Parallel Computer Architecture: A Hardware/Software Approach

Prerequisites

- Linux
- C/Fortran Programming
- Computer Architecture
- Operating Systems
- Numerical Methods

Exams

- Attendance of lectures – 10%
- Assignments - 30%
- Project - 30%
- Exam – 30%

Course Roadmap

□ 博士生实践课
文本 本科生课程

本科课时	博士课时	内容	
2	2	并行处理基础	
2	2	理解并行、延迟和带宽	多核体系结构
2	2	并行和通信的方式	共享内存并行、消息传递并行、数据并行
8	8	如何编写和优化并行程序	并行编程基础 → 任务分配和调度 → 局部性、通信及竞争 → GPU结构及编程
2	4	并行程序案例学习和分析	稠密线性代数 → SIMD
4	8	系统性能评价方法	性能模型、模拟器 → 基准测试程序（• PARSEC • HPL • HPCG • Graph500） → Roofline模型
4	8	共享内存结构及编程	相关性协议（• Snoop • Diretory） → 一致性模型 → OpenMP编程
10	6	消息传递结构及编程	RDMA硬件 → IB verbs编程 → MPI编程
2	4	互连网络设计与实现	系统互连网络拓扑结构 → 大规模互连网络模拟
0	6	同步操作设计与实现	锁、栅障的设计与实现 → 事务性内存结构 → 无锁算法实践
0	4	并行文件系统	并行文件系统的硬件、系统设计、案例 → 并行I/O编程
0	2	数据仓储级并行系统	数据中心、云计算的扩展性
0	2	异构并行和领域专用加速	软件定义硬件体系结构

备注：本科生2个大作业课时、2个期末考试课时

Course Project

- **基本要求**
  - 成功运行所有并行程序，并记录实验时间
  - 制作报告讲解并行算法，并演示程序运行
- **必选题**
  - HPL: http://www.netlib.org/benchmark/hpl/
  - HPCG: http://hpcg-benchmark.org
  - Graph500: https://graph500.org
  - 根据调优的性能排序额外增加分数
- **附加题**
  - 自己选择一个算法，实现其并行算法

"It's not a human move ..."

- We experience seemingly daily new milestones in machine intelligence:
- Alpha Go of Deep Mind (Google) winning Go against Lee Sedol, one of the world's top go players, March 11, 2016.
- Alpha Go Zero: "Mastering the Game of Go without Human Knowledge", learning only from self-play, Nature 550, October 19, 2017

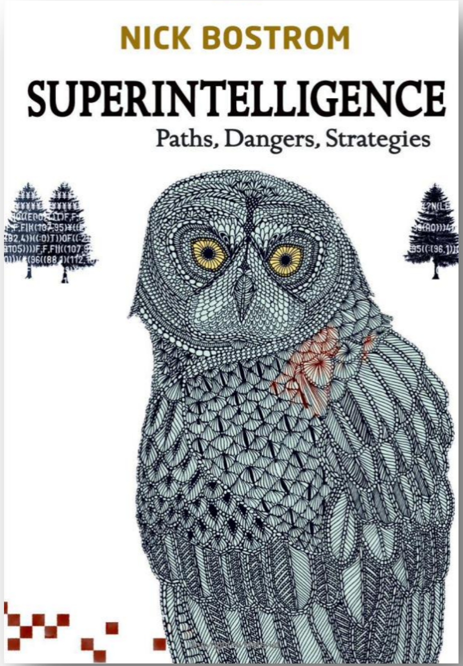
http://www.wired.com/2016/03/sadness-beauty-watching-googles-ai-play-go/?mbid=social_fb

Recent concerns about the "machines" taking over

"Those disposed to dismiss an 'AI takeover' as science fiction may think again after reading this original and well-argued book."
– Martin Rees, Past President, Royal Society

"Worth reading *Superintelligence* by Bostrom. We need to be super careful with AI. Potentially more dangerous than nukes."
– Elon Musk

"If our own extinction is a likely, or even possible, outcome of our technological development, shouldn't we proceed with great caution?"
– Bill Joy



"Success in creating AI would be the biggest event in human history. Unfortunately, it might also be the last, unless we learn how to avoid the risks."
– Steven Hawking

Slide adapted from Larry Smarr, UCSD

And what about supercomputers?

- At any given time one of the most powerful computers to solve scientific and engineering problems
- Supercomputers and HPC are largely absent from the public discussion about progress in AI

Outline

- A Brief History of Supercomputer
- Why all computers must be parallel processors
  - Including your laptops and handhelds
- Large Computational Science and Engineering (CSE) problems require powerful computers
  - Commercial problems too
- Why writing (fast) parallel programs is hard
  - But things are improving

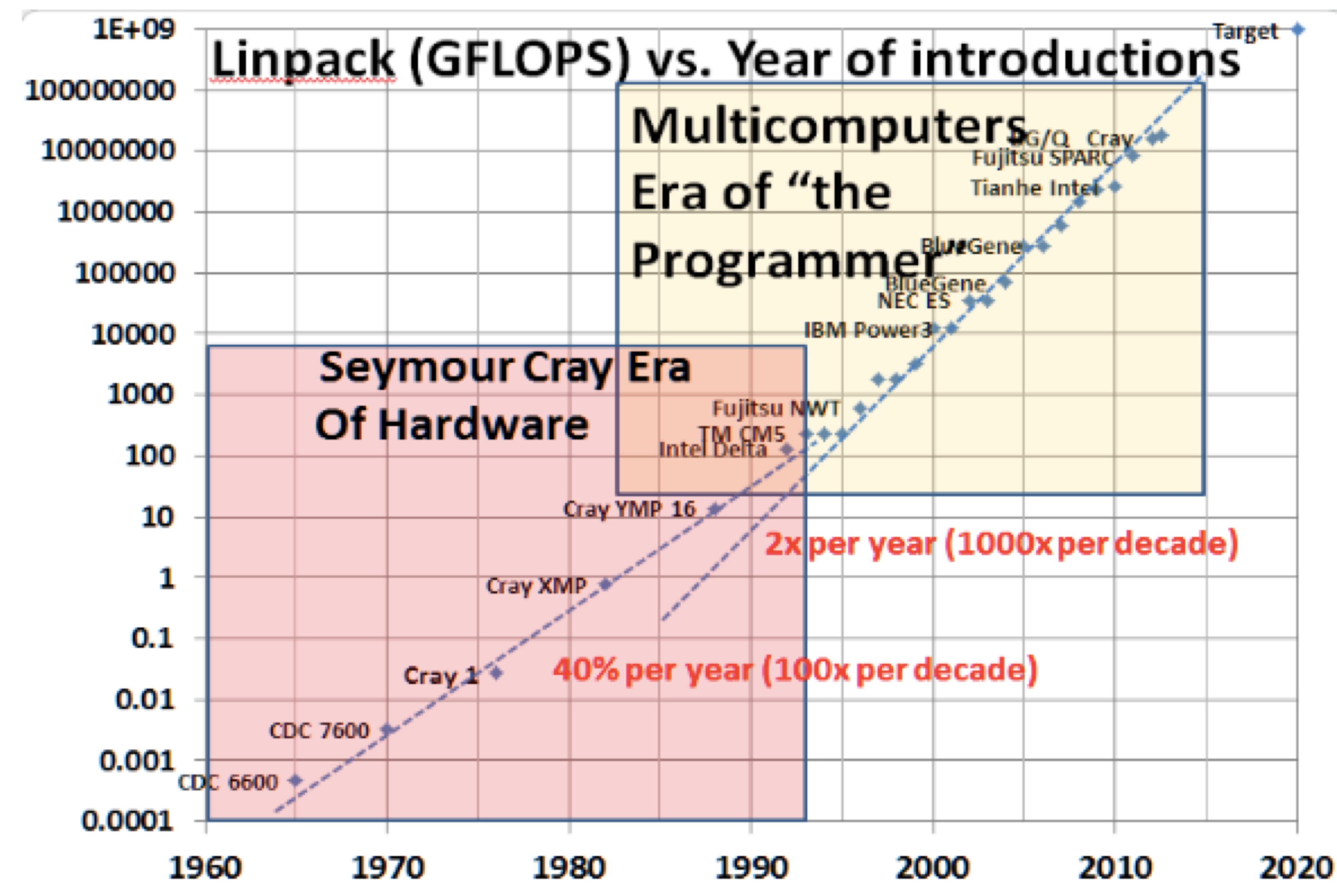
Unites of Measure

- High Performance Computing (HPC) units are:
  - Flop: floating point operation, usually double precision unless noted
  - Flop/s: floating point operations per second
  - Bytes: size of data (a double precision floating point number is 8 bytes)
- Typical sizes are millions, billions, trillions…

Mega	Mflop/s = 10^6 flop/sec	Mbyte = 2^{20} = 1048576 ~ 10^6 bytes
Giga	Gflop/s = 10^9 flop/sec	Gbyte = 2^{30} ~ 10^9 bytes
Tera	Tflop/s = 10^{12} flop/sec	Tbyte = 2^{40} ~ 10^{12} bytes
Peta	Pflop/s = 10^{15} flop/sec	Pbyte = 2^{50} ~ 10^{15} bytes
Exa	Eflop/s = 10^{18} flop/sec	Ebyte = 2^{60} ~ 10^{18} bytes
Zetta	Zflop/s = 10^{21} flop/sec	Zbyte = 2^{70} ~ 10^{21} bytes
Yotta	Yflop/s = 10^{24} flop/sec	Ybyte = 2^{80} ~ 10^{24} bytes

- Current fastest (public) machine 143.5 Pflop/s, 2,397,824 cores
  - Up-to-date list at www.top500.org

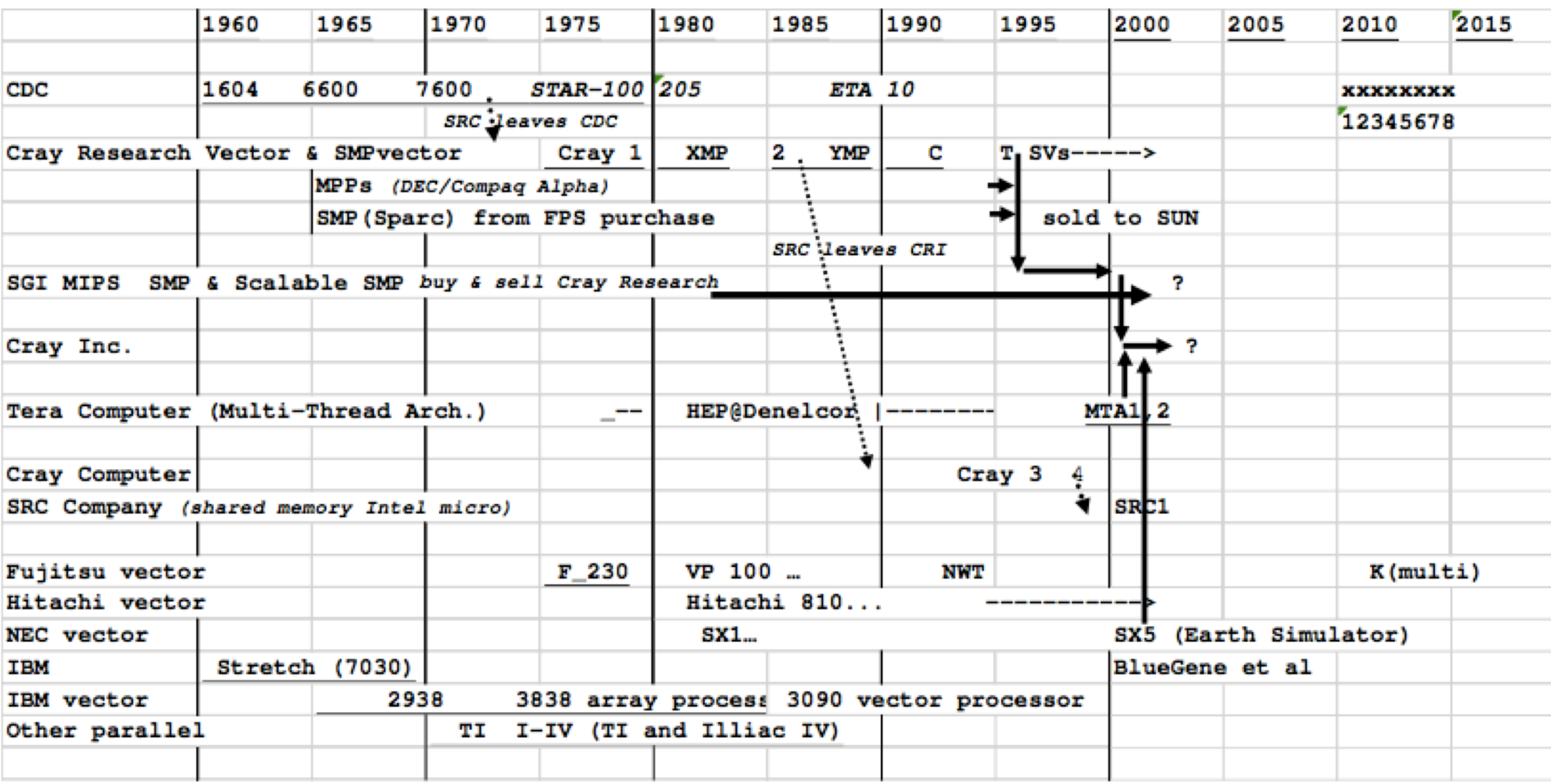
Two Eras of Supercomputers



Two Eras of Supercomputers

	Cray Era (Monocomputer)	Multicomputer Era
Characterization	One, shared memory, accessed by one or more processors	Multiple independent computers that are interconnected via one or more high speed networks
Date	1964-1993	1983-present
Defining Systems	CDC 6600, CDC 7600, CDC Star, Cray 1 … C90, NEC SX	Caltech Cosmic Cube, Thinking Machines CM5, ASCI Red, …Tianh2
Challenge	Hardware and architecture. Semiconductors. Processor parallelism, memory latency & bandwidth. Multiprocessing and multithreading	Programming a multicomputer. Fast processors and processing elements that form each node. Fast and low latency network interconnections.
Standards	FORTRAN (single memory)	MPI; Beowulf
Gain: Linpack Gain: Clock Gain: Explicitly controlled P's	 10 MHz-500 MHz 13% 64 P	 5 MHz-5 GHz 25%/yr 64-3 million P&PE 43%/yr

Monocomputer Era (1964-1993)



The Multicomputer Era (1993-Present)

The CMOS Goal Rush: The transition to multicomputers (1983-1993)

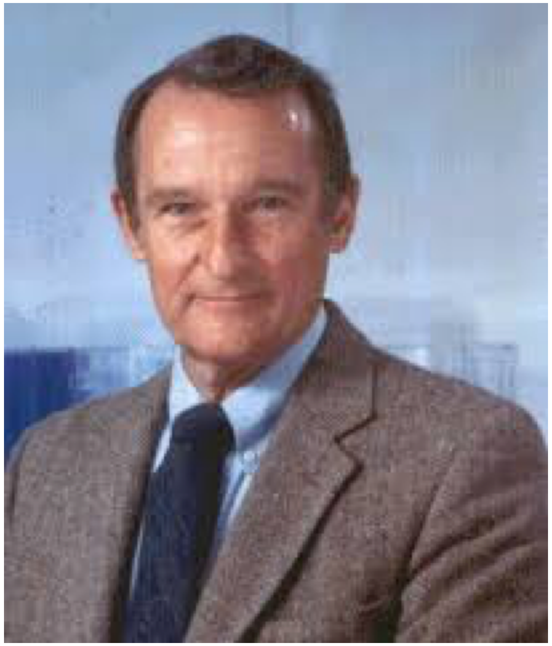
Yr.	Rmax(Gflops)	Rpeak(Gflops)	Cores	Kwatt	Computer
1982	**1**	**1**	**4**		**Cray XMP mPv**
1983		?	4>64		Cosmic Cube @Caltech
1987		?	1024		nCUBE @Sandia Nat. Lab.
1988	14	15.2	16		Cray YMP 16 (C90) mPvect
1993-4		20	512		Intel Delta/Paragon @Caltech
1995		64	32		Cray T90 32 mPvect
	Top500 Listing of computers started June 1993				
1993.6	60	131	1024		TM CM5 @LANL
1993.9	124	236	140		Fujitsu mCvect @NWT
1994	124	236	140		
1995	124	236	140		
1996	368	614	1024		Hitachi mCvect @Tokyo U.
1997	**1,338**	**1,830**	**9,632**		**ASCI Red Intel @Sandia**
1998	1,338	1,830	9,632		
1999	2,400	3,200	9,632	850	(upgraded)
2000	4,938	12,288	8,192		ASCI White IBM SP3 @LLNL
2001	4,938	12,288	8,192		IBM Power3
2002	35,860	40,960	5,120	3,200	NEC Earth Simulator mCvect.
2003	35,680	40,960	5,120	3,200	(June 2002-June 2004)
2004	70,720	91,750	32,768		IBM BlueGene/L @LLNL
2005	281,000	367,000	131,072	1,433	
2006	281,000	367,000	131,072	1,433	
2007	478,200	596,000	212,992	2,329	
2008	**1,105,000**	**1,456,000**	**129,600**	**2,483**	**IBM Roadrunner Blade @LANL**
2009	1,759,000	2,331,000	224,000	6,950	Cray Jaguar 6 core Intel @ORNL
2010	2,567,000	4,701,000	186,386	4,040	Tianhe Intel @China NUDT
2011	10,510,000	11,280,000	705,024	12,659	Fujitsu SPARC
2012	16,325,000	20,132,000	560,640	8,209	Cray XK7 Intel & GPU @ORNL
2013	33,860,000	54,902,000	3,120,000	17,808	Tianhe2 Intel @China NUDT

Seymour Roger Cray

- The father of supercomputing

- CDC 6600/7600/8600
- Cray-1/2/3/4

- Control Data Corporation
- Cray Research/Computer Corporation
- SRC Computers



September 28, 1925– October 5, 1996

"Anyone can build a fast CPU. The trick is to build a fast system."

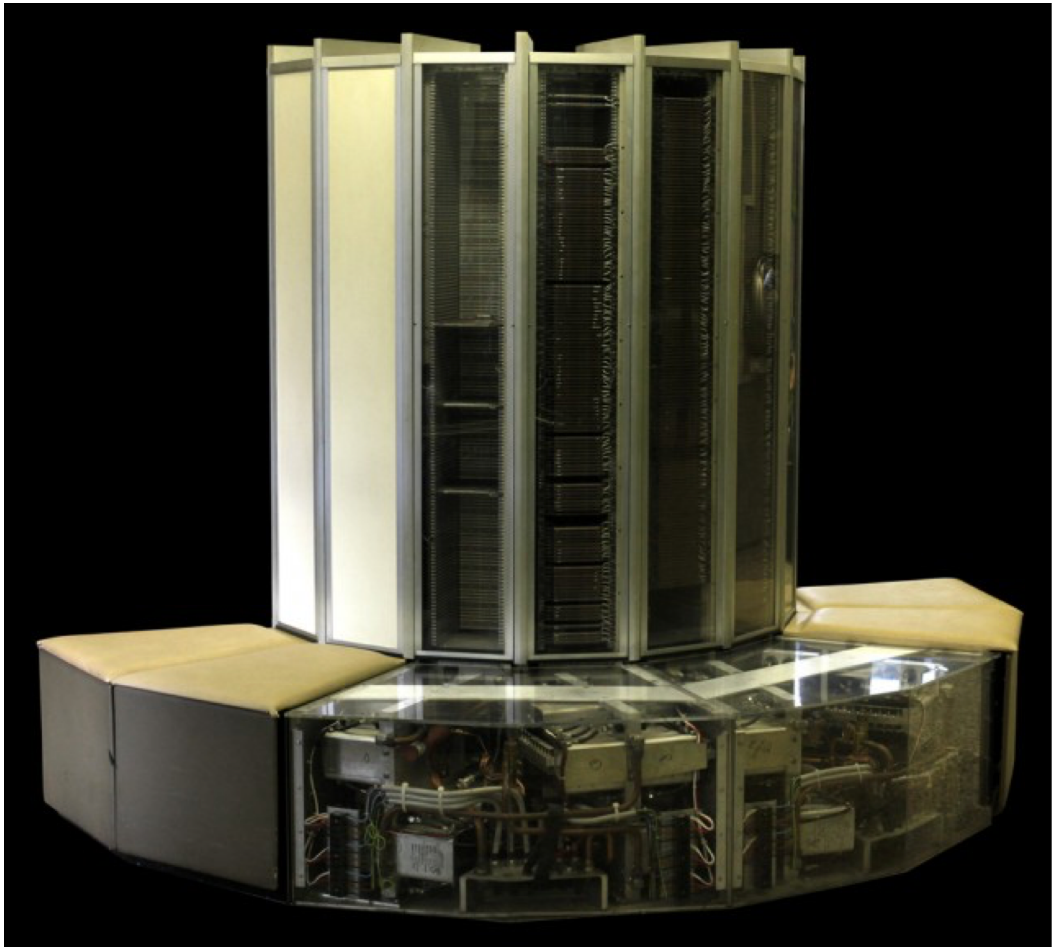
The First Supercomputer [1964]

- CDC 6600
- 3 million floating point operations per second (flops)
- 1CPU, 40MHz
  - 10 parallel functional units, each of which were dedicated to different tasks; floating point add, floating point divide, boolean logic, etc.
  - 60-bit word length and 60-bit registers
  - The first RISC system
- $13M

- 10 times faster than the fastest computer (IBM 7030 Strech) at the time
- Cooled with Freon that circulated in pipes around the four cabinets
- 10 Peripheral Processors, each of which was dedicated to managing I/O and keeping the CPU's queue full

Vector Supercomputer [1976]

- Cray 1
- 136 megaflops
- 1,662 printed circuit boards with up to 144 ICs on each
  - eight 64-element by 64-bit vector (V) registers, as well as a vector length (VL) and vector mask (VM)
- 80MHz
- $8 million

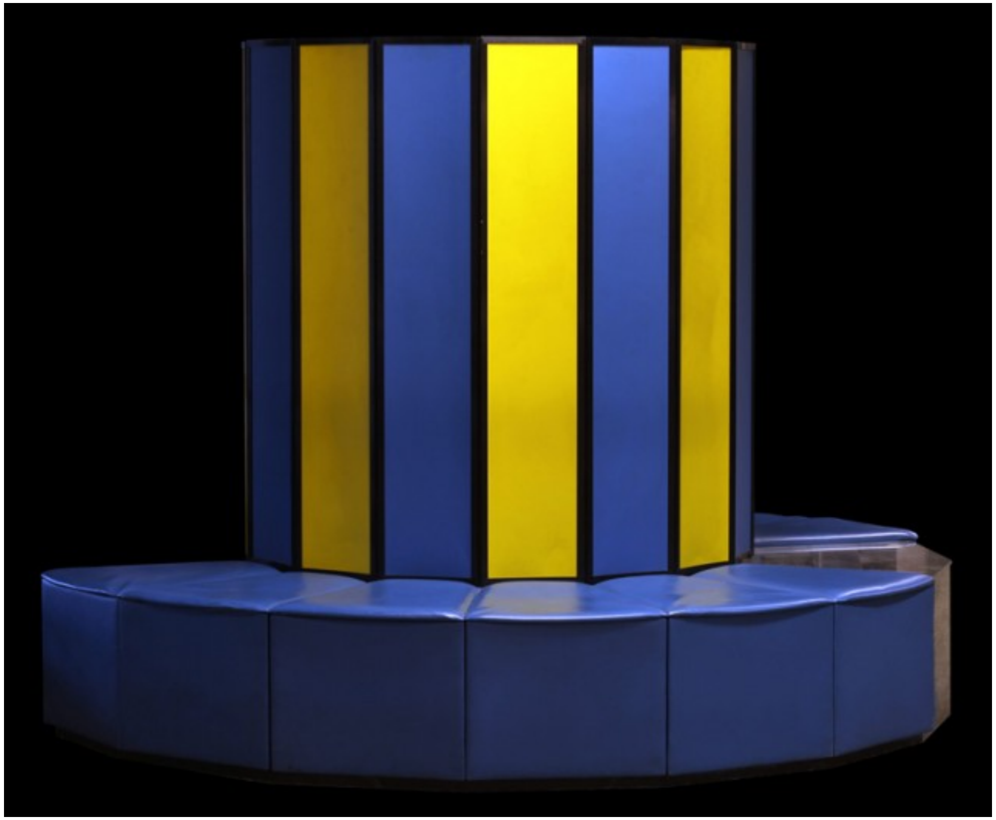


Installed at Los Alamos National Laboratory


SC
Conference Series
30TH ANNIVERSARY
1988–2018
CRAY-1
CRAY RESEARCH, INC.
1995

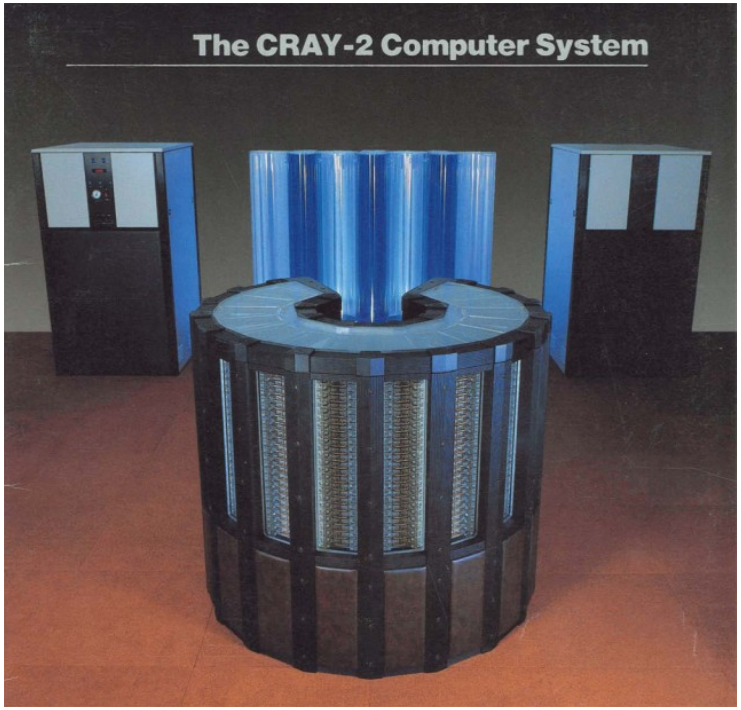
Multi-CPU Supercomputer [1982]

- Cray-XMP
- 800 megaflops
- Four CPUs, 105MHz
  - 16 million 64-bit words of SRAM memory
  - 32 disk storage units
  - 1.2GB, 10MB/s
- $15M



The First Commodity Supercomputer [1985]

- Cray 2
- 1.9 gigaflops
- "foreground" processors to load data from main memory to local memory (similar to a cache but not quite) via a very fast gigabit-per-second bus, and then pass instructions off to "background" processors which would actually perform computation
- Run "mainstream" software, thanks to UniCOS, a Unix System V derivative with some BSD features
- Used in many universities and corporations



the entire computer was submersed in Fluorinert (全氟三丁胺)

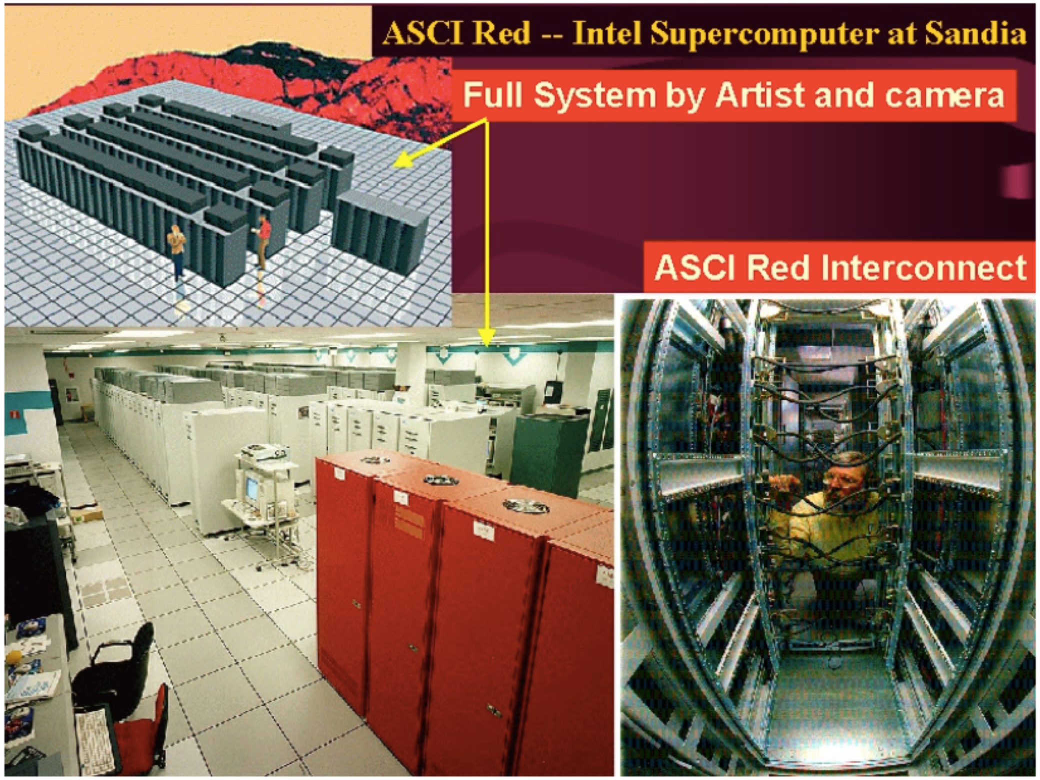
Massive Parallel Processing [1996]

- Hitachi SR2201
- 600 gigaflops
- Highly parallel arrays of vector processors attached to fast memory
- 2D and 3D networks (such as Cray's Torus interconnect) connected together hundreds of CPUs

Off-the-shelf CPUs Supercomputer [1996]

The first supercomputer to reach a Teraflops and made from off-the-shelf CPUs — Pentium Pros, and then Pentium II Xeons — and other readily-available commercial components

- ASCI Red
- 1.3Tflops
- 6,000 200MHz Intel Pentium Pros CPUs
- A distributed memory MIMD (Multiple Instruction, Multiple Data) message-passing computer
- $46 million



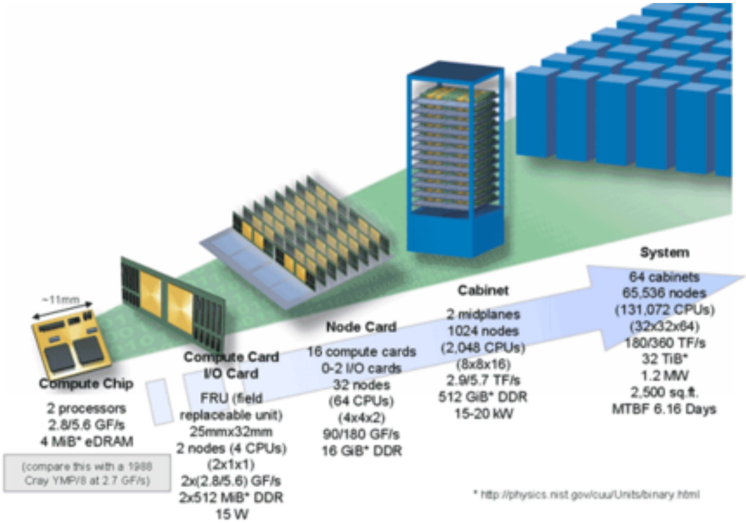
Sandia National Laboratories

The REAL Commodity Supercomputer [1998]

- Beowulf cluster
- Cluster of Workstations (COW)
- MPI
- PVM
- Linux

Low Power Supercomputer [2004-2008]

- BlueGene/L/P/Q
- 16,000 compute nodes (each with two CPUs) and was capable of 70 teraflops (2004) → more than 100,000 compute nodes and peak performance of 600 teraflops (2007)
- except for RAM, the compute nodes were entirely integrated into SoCs (system-on-a-chip)



The development of ever-more-complex interconnects, and reducing power usage

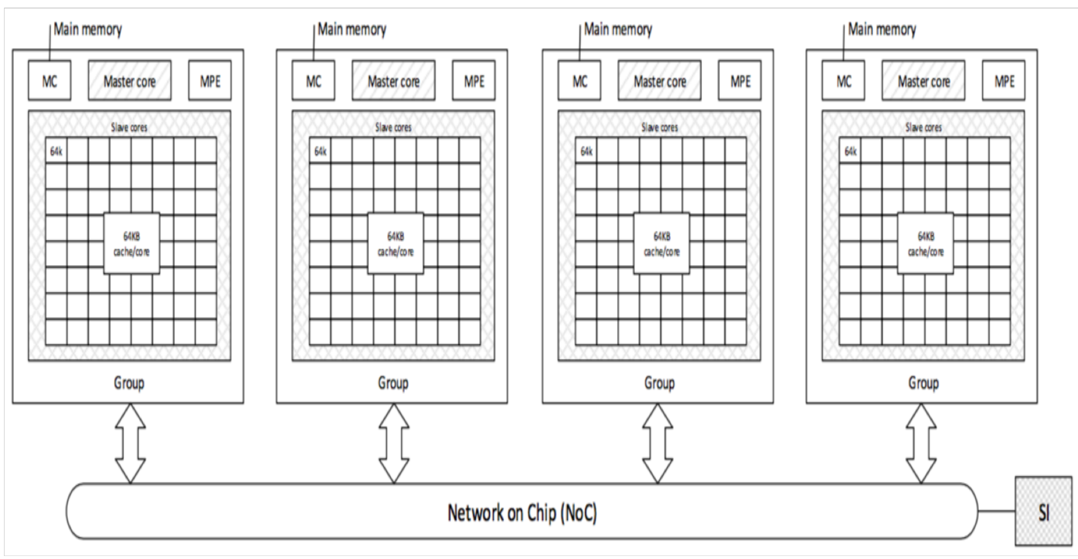
Accelerator Supercomputer [2008]

- IBM Roadrunner
- 1.026 Petaflops
- 6,912 Opteron processors
- 12,960 PowerXCell processors (accelerators)

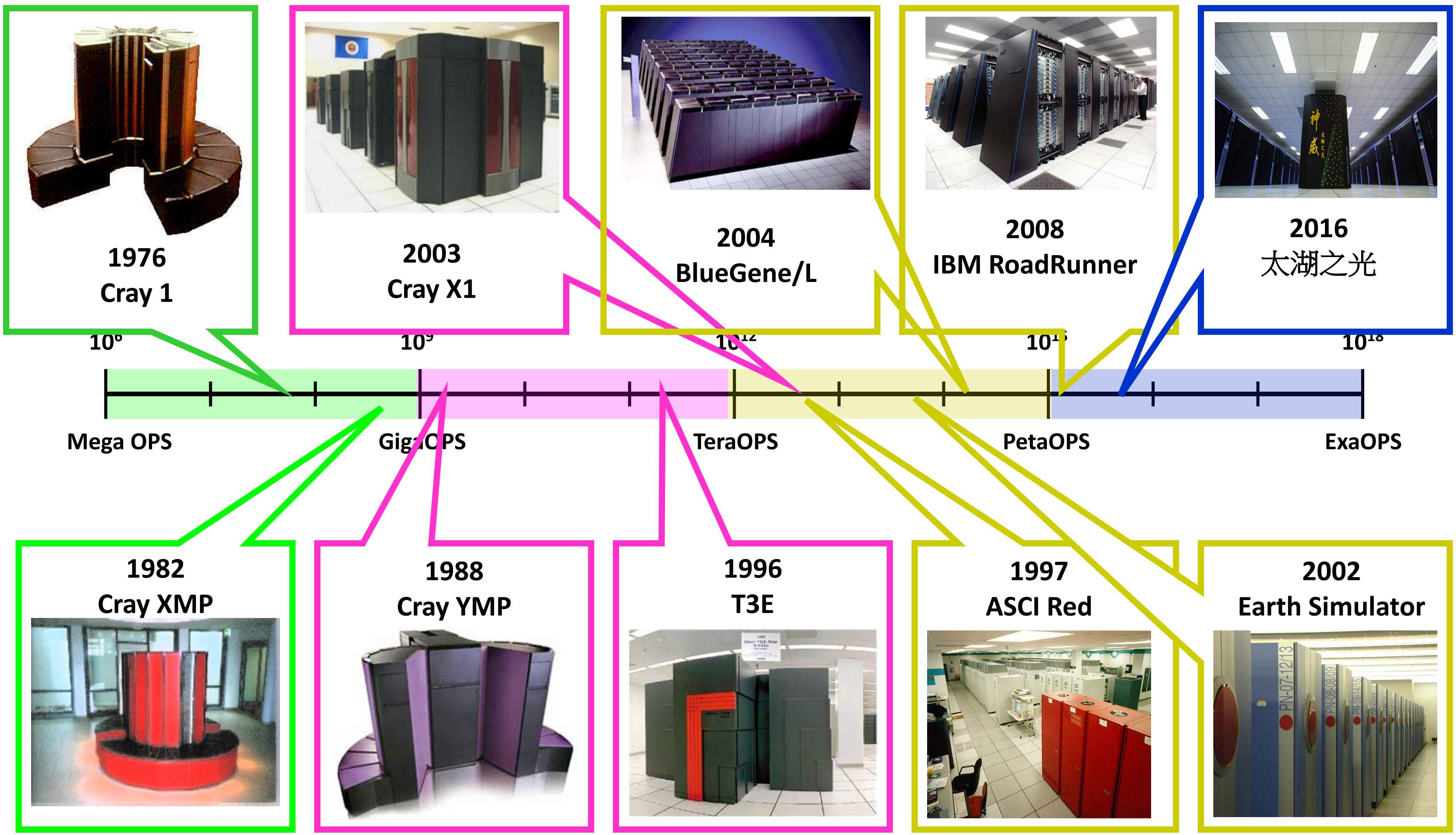
The first computer to break the 1-petaflop barrier

Manycore Supercomputer [2016]

- 神威太湖之光
- 125.4Pflops
- The processor chip is composed of 4 core groups (CGs), connected via a NoC,each of which includes a Management Processing Element (MPE) and 64 Computing Processing Elements (CPEs) arranged in an 8 by 8 grid



A Growth-Factor of a Billion in Performance



中科院计算所超算之路

1956年 1990年 2000年

1958年
我国第一台电子计算机103机

1967年
“两弹一星”功勋机
109丙机


夏培肃


高庆狮

1990年
启动高性能计算机研制

1990~1997
曙光MPP：
曙光一号，
曙光1000

1998~2013
曙光机群：
曙光2000，曙光3000，
曙光4000，曙光5000，曙光6000

2015年
曙光公司成功上市

研制曙光高性能计算机，打破IBM市场垄断，实现高端装备规模产业化

曙光高性能计算机

曙光1号

全对称共享存储
多处理机系统

曙光1000

当时中国最快的计算机
峰值25.6亿次/秒

曙光2000

中国首个机群系统
峰值1117亿次/秒

曙光3000

工业标准机群
峰值4032亿次/秒

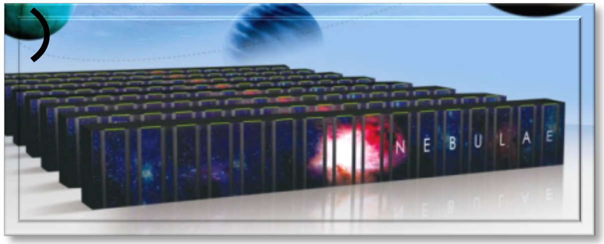
曙光4000

中国首个进入TOP500
前十名的高性能计算机、
峰值11万亿次/秒

曙 光5000

当时中国最快的计算机、
中国首个刀片式机群、
峰值230万亿次/秒

曙光6000（星云）



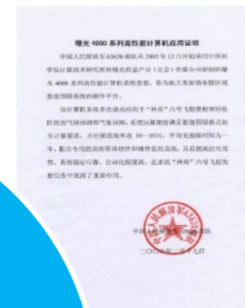
中国首个实测性能超过千万亿
次的高性能计算机、
世界TOP500第二名、
峰值3千万亿次/秒

打破国际巨头对国家重要信息基础设施的垄断

本世纪初70%用曙光机

随着BGP的102支物探队走向世界

神舟系列飞船发射保障



曙光4000系列高性能计算机应用证明

中国人民解放军63620部队从2003年12月开始采用中国科学院计算技术研究所和曙光信息产业（北京）有限公司研制的曙光4000系列高性能计算机系统壹套，作为航天发射场有限区域数值预报系统的硬件平台。

该计算机系统多次成功应用于"神舟"六号飞船发射和回收阶段的气候预测和气象保障，系统运算速度满足数值预报模式积分计算要求，并行算法效率在80—90%，平均无故障时间为一年。配合专用的系统管理软件和硬件监控系统，具有很高的可用性。系统稳定可靠，自动化程度高，在承担"神舟"六号飞船发射任务中发挥了重要作用。

中国人民解放军63620部队

二〇〇[illegible]年二月十七日

西安卫星发射中心

酒泉卫星发射中心

北京卫星控制指挥中心

国家天文台

紫金山天文台

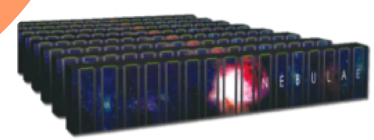
国家级信息关防
国家级短信平台

阻断有害信息访问超1000亿次

提供上千万份情报信息

国家级超算中心

上海、深圳



最大的工业创新平台

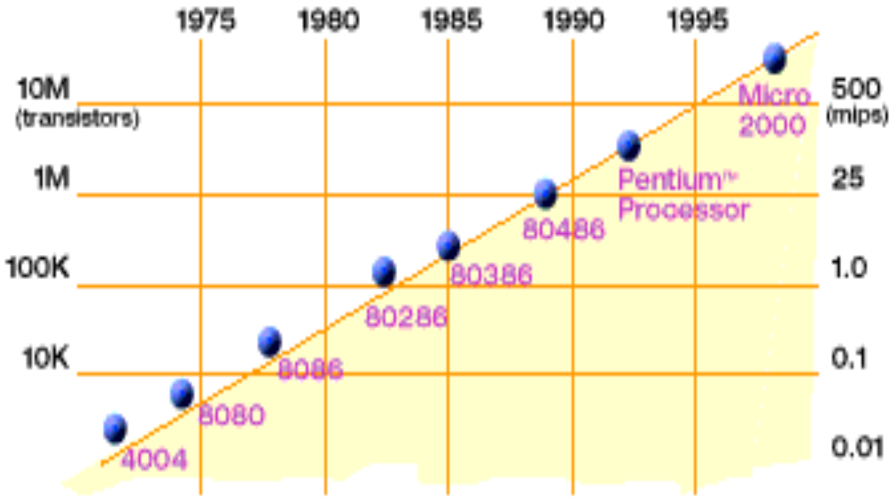
南京等12个城市云计算中心

Why powerful computers are parallel

Tunnel Vision by Experts

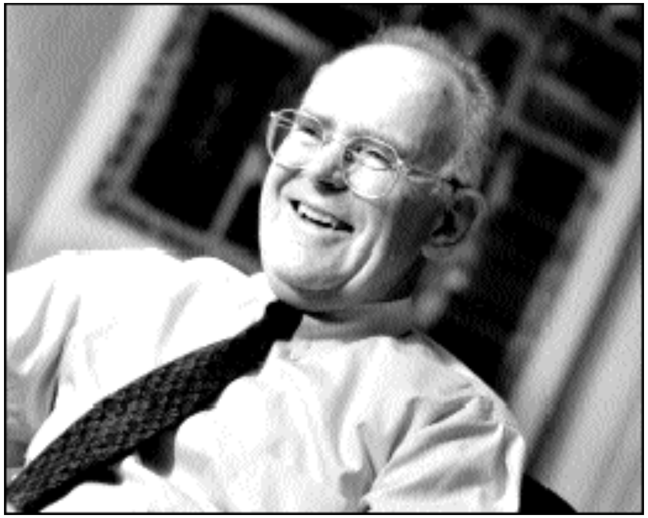
- "I think there is a world market for maybe five computers."
  - Thomas Watson, chairman of IBM, 1943.
- "There is no reason for any individual to have a computer in their home"
  - Ken Olson, president and founder of Digital Equipment Corporation, 1977.
- "640K [of memory] ought to be enough for anybody."
  - Bill Gates, chairman of Microsoft,1981.
- "On several recent occasions, I have been asked whether parallel computing will soon be relegated to the trash heap reserved for promising technologies that never quite make it."
  - Ken Kennedy, CRPC Directory, 1994

Technology Trends: Microprocessor Capacity



2X transistors/Chip Every 1.5 years

Called "Moore's Law"



Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months.

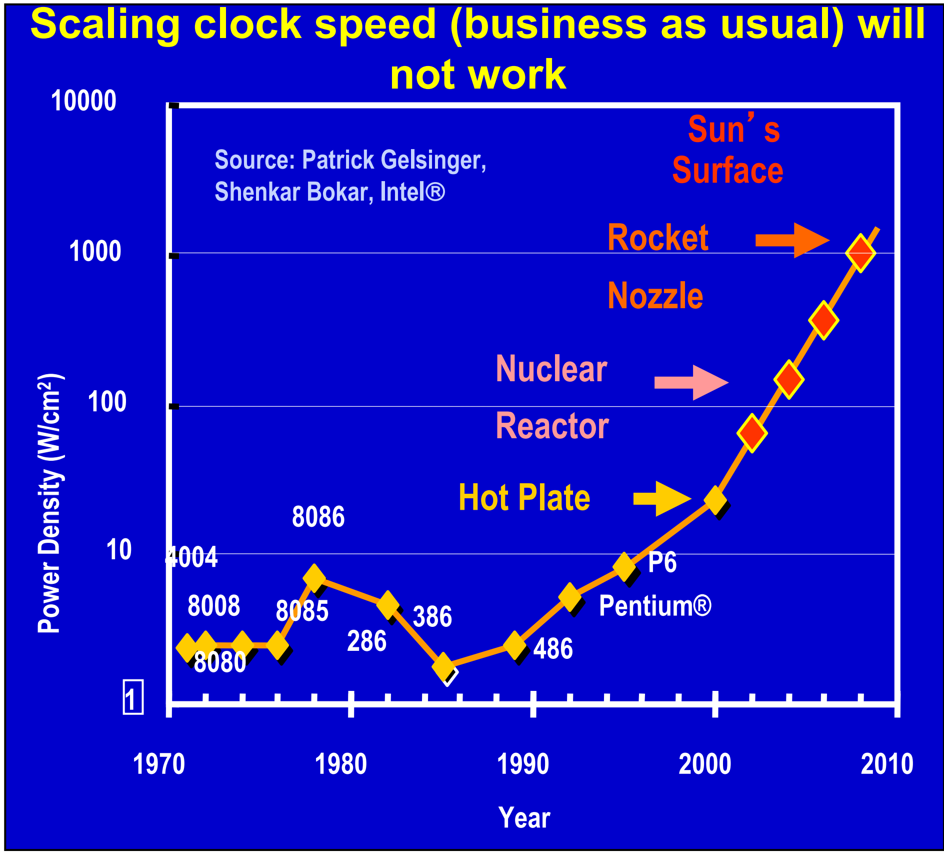
- Microprocessors have become smaller, denser, and more powerful.

Impact of Device Shrinkage

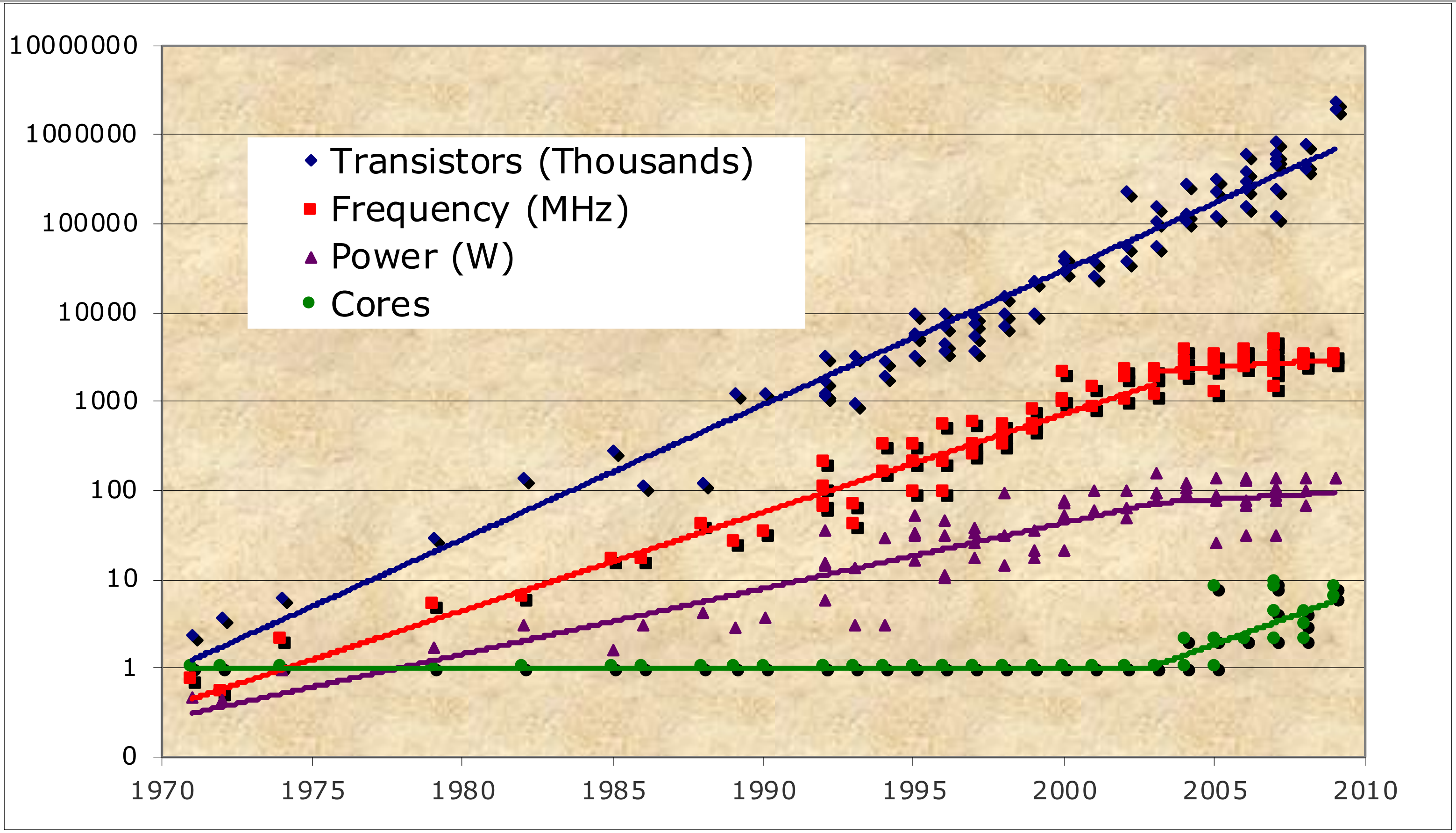
- What happens when the feature size (transistor size) shrinks by a factor of x ?
- Clock rate goes up by x because wires are shorter
  - actually less than x, because of power consumption
- Transistors per unit area goes up by x^2
- Die size also tends to increase
  - typically another factor of ~x
- Raw computing power of the chip goes up by ~ x^4 !
  - typically x^3 is devoted to either on-chip
    - parallelism: hidden parallelism such as ILP
    - locality: caches
- So most programs x^3 times faster, without changing them

Power Density Limits Serial Performance

- **Concurrent systems are more power efficient**
  - Dynamic power is proportional to V^2fC
  - Increasing frequency (f) also increases supply voltage (V) → cubic effect
  - Increasing cores increases capacitance (C) but only linearly
  - Save power by lowering clock speed
- **High performance serial processors waste power**
  - Speculation, dynamic dependence checking, etc. burn power
  - Implicit parallelism discovery
- **More transistors, but not faster serial processors**



Revolution in Processors



- Chip density is continuing increase ~2x every 2 years
- Clock speed is not
- Number of processor cores may double instead
- Power is under control, no longer growing

Moore’s Law reinterpreted

- Number of cores per chip can double every two years
- Clock speed will not increase (possibly decrease)
- Need to deal with systems with millions of concurrent threads
- Need to deal with inter-chip parallelism as well as intra-chip parallelism

Parallelism in 2017?

- **These arguments are no longer theoretical**
- **All major processor vendors are producing *multicore* chips**
  - Every machine will soon be a parallel machine
  - To keep doubling performance, parallelism must double
- **Which (commercial) applications can use this parallelism?**
  - Do they have to be rewritten from scratch?
- **Will all programmers have to be parallel programmers?**
  - New software model needed
  - Try to hide complexity from most programmers – eventually
  - In the meantime, need to understand it
- **Computer industry betting on this big change, but does not have all the answers**
  - YOU!

The TOP500 Project

- Listing the 500 most powerful computers in the world
- Yardstick: Rmax of Linpack
  - Solve Ax=b, dense problem, matrix is random
  - Dominated by dense matrix-matrix multiply
- Updated twice a year:
  - ISC'xy in June in Germany
  - SCxy in November in the U.S.
- All information available from the TOP500 web site at: www.top500.org

Outline

- Why powerful computers must be parallel processors
  - Including your laptops and handhelds
- Large CSE problems require powerful computers
  - Commercial problems too
- Why writing (fast) parallel programs is hard
  - But things are improving

Computational Science - News

"An important development in sciences is occurring at the intersection of computer science and the sciences that has the potential to have a profound impact on science. It is a leap from the application of computing … to the *integration of computer science concepts, tools, and theorems* into the very fabric of science." -*Science* 2020 Report, March 2006

Nature, March 23, 2006

Simulation in Science and Engineering

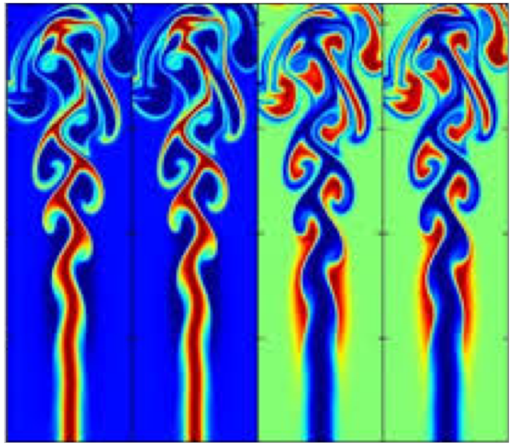
High performance computing (HPC) simulation to understand things that are:

- too big
- too small
- too fast
- too slow
- too expensive or
- too dangerous

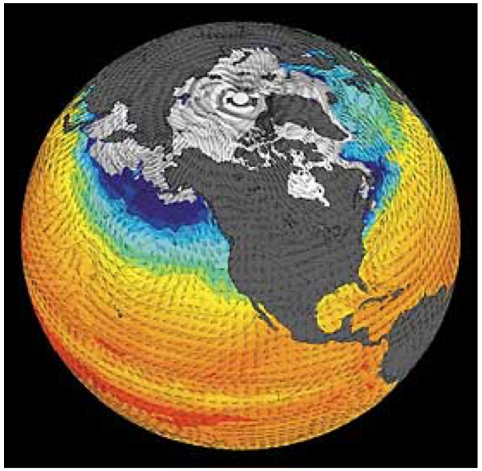
for experiments

Understanding the universe

Proteins and diseases

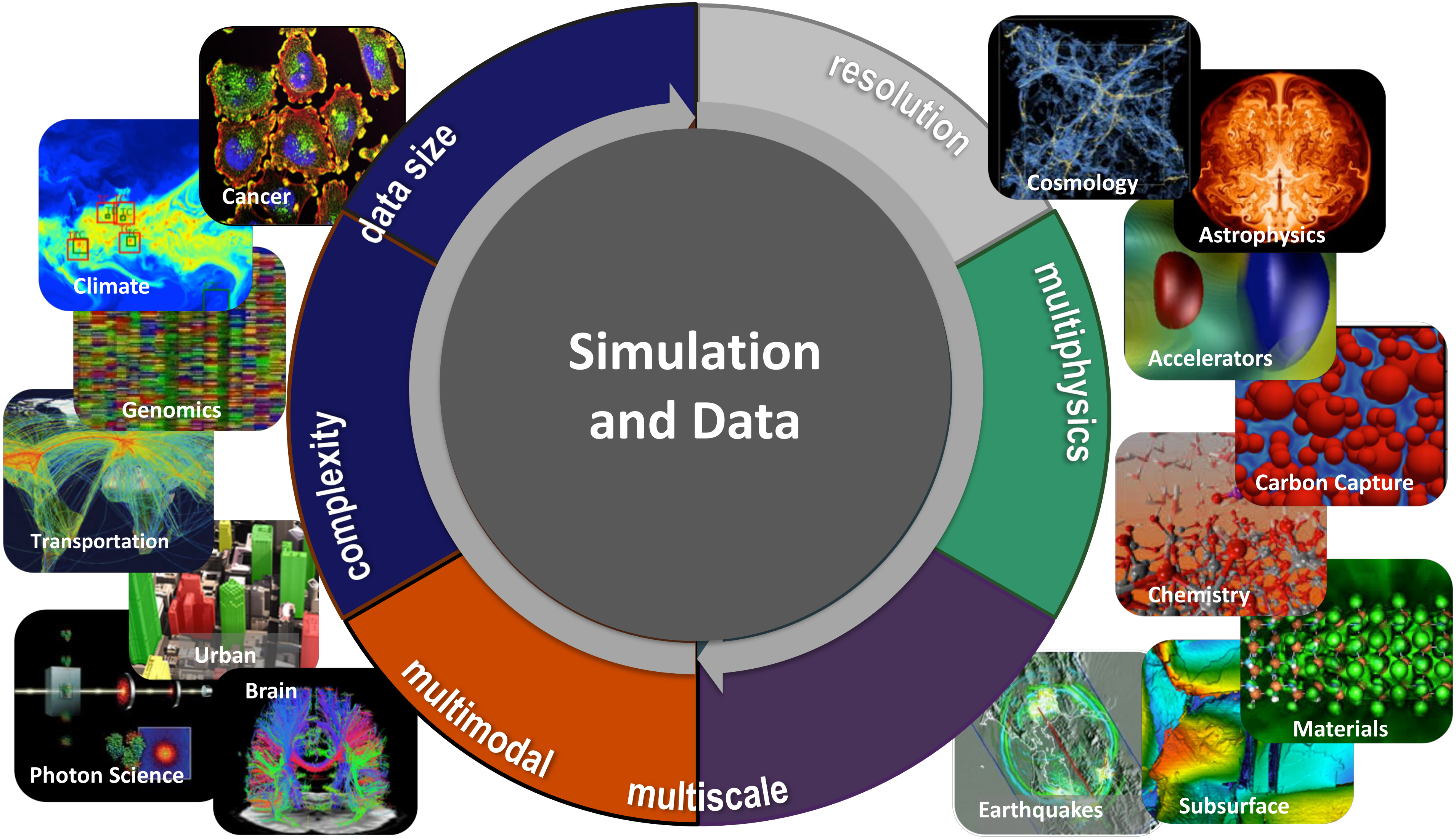


Energy-efficient jet engines



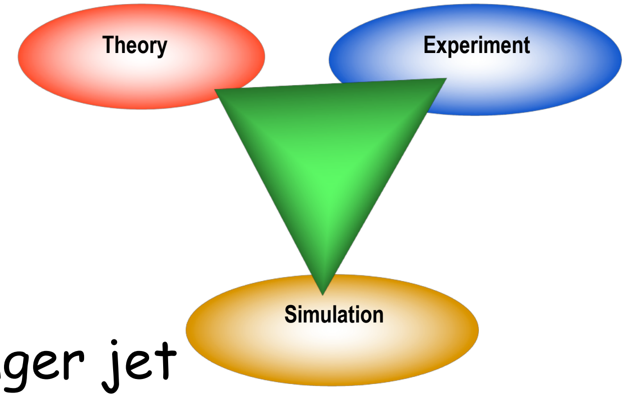
Climate change

Breakthrough Science at the Exascale will combine simulation and data analysis



Simulation: The Third Pillar of Science

- **Traditional scientific and engineering method:**
  - (1) Do theory or paper design
  - (2) Perform experiments or build system



- **Limitations:**
  - -Too difficult—build large wind tunnels
  - -Too expensive—build a throw-away passenger jet
  - -Too slow—wait for climate or galactic evolution
  - -Too dangerous—weapons, drug design, climate experimentation
- **Computational science and engineering paradigm:**
  - (3) Use computers to simulate and analyze the phenomenon
  - Based on known physical laws and efficient numerical methods
  - Analyze simulation results with computational tools and methods beyond what is possible manually

Class A and Class B Applications

- Class A
  - Applications that are highly parallelizable
  - Low in communication
  - Regular
  - Logic flow depends little on the input data
  - Examples: Matrix Multiply, dot product, image processing (i.e. JPEG decompressing and compressing), etc
- Class B
  - Applications in which parallelism is hidden or non existent
  - High Communication / Synchronization needs
  - Logic flows depends greatly on input data (while loops, conditional structures)
  - Examples: Sorting methods, graph and tree searching

Science Grand Challenges

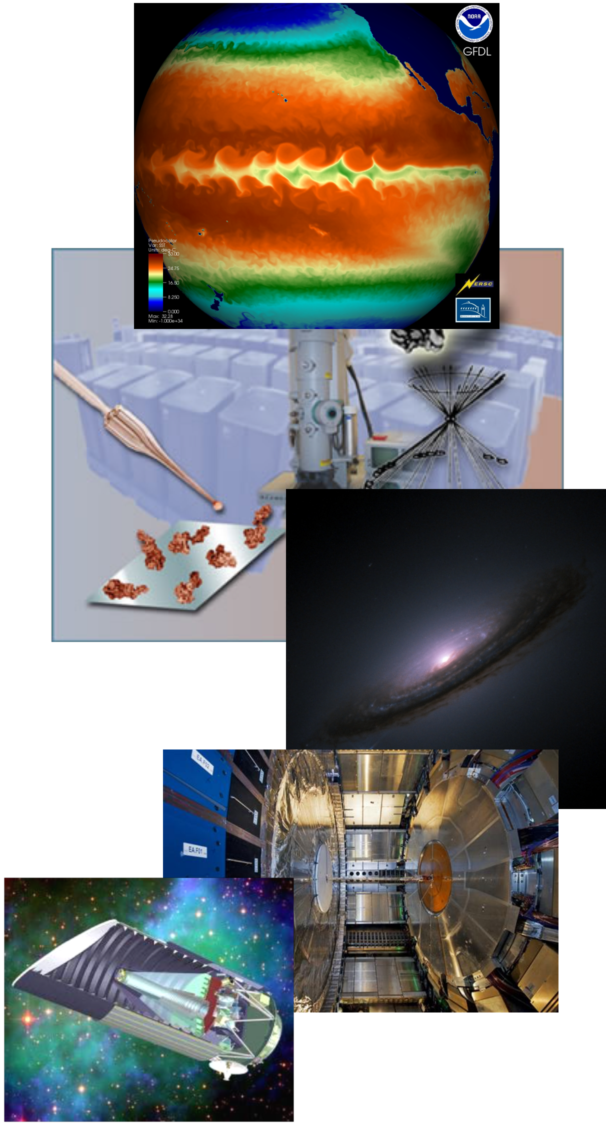
- Quantum Chemistry, Statistical Mechanics and relativistic physics
- Cosmology and astrophysics
- Computational Fluid dynamic and turbulence
- Material Design and Superconductivity
- Biology, Pharmacology, genome sequencing, genetic engineering, protein folding, enzyme activity, and cell modeling
- Medicine and modeling of human organs and bones
- Global weather and environmental modeling

Drivers for Change

- **Continued exponential increase in computational power**
  - Can simulate what theory and experiment can't do
- **Continued exponential increase in experimental data**
  - Moore's Law applies to sensors too
  - Need to analyze all that data

Data Driven Science

- **Scientific data sets are growing exponentially**
  - Ability to generate data is exceeding our ability to store and analyze
  - Simulation systems and some observational devices grow in capability with Moore's Law
- **Petabyte (PB) data sets will soon be common:**
  - Climate modeling: estimates of the next IPCC data is in 10s of petabytes
  - Genome: JGI alone will have .5 petabyte of data this year and double each year
  - Particle physics: LHC is projected to produce 16 petabytes of data per year
  - Astrophysics: LSST and others will produce 5 petabytes/year (via 3.2 Gigapixel camera)
- **Create scientific communities with "Science Gateways" to data**



Some Particularly Challenging Computations

- **Science**
  - Global climate modeling
  - Biology: genomics; protein folding; drug design
  - Astrophysical modeling
  - Computational Chemistry
  - Computational Material Sciences and Nanosciences
- **Engineering**
  - Semiconductor design
  - Earthquake and structural modeling
  - Computation fluid dynamics (airplane design)
  - Combustion (engine design)
  - Crash simulation
- **Business**
  - Financial and economic modeling
  - Transaction processing, web services and search engines
- **Defense**
  - Nuclear weapons -- test by simulations
  - Cryptography

Economic Impact of HPC

- **Airlines:**
  - System-wide logistics optimization systems on parallel systems.
  - Savings: approx. $100 million per airline per year.
- **Automotive design:**
  - Major automotive companies use large systems (500+ CPUs) for:
    - CAD-CAM, crash testing, structural integrity and aerodynamics.
    - One company has 500+ CPU parallel system.
  - Savings: approx. $1 billion per company per year.
- **Semiconductor industry:**
  - Semiconductor firms use large systems (500+ CPUs) for
    - device electronics simulation and logic validation
  - Savings: approx. $1 billion per company per year.
- **Energy**
  - Computational modeling improved performance of current nuclear power plants, equivalent to building two new power plants.

超级计算的重要性

- 在美国**PITAC**重点支持的四个方向列首位:
  - High End Computing and Computation
    - HEC Infrastructure and Application
    - HEC Research and Development
  - Human Computer Interface and Informati Management
  - Large Scale Networking
  - Software Design and Productivity
- **ASCI**计划，**ASCR**计划及**FastForward2 Project**
  - 核武库世界领先地位与安全确保计划
- 高端计算需求始终是计算机发展的火车头

What Supercomputers Do

Global Climate Modeling Problem

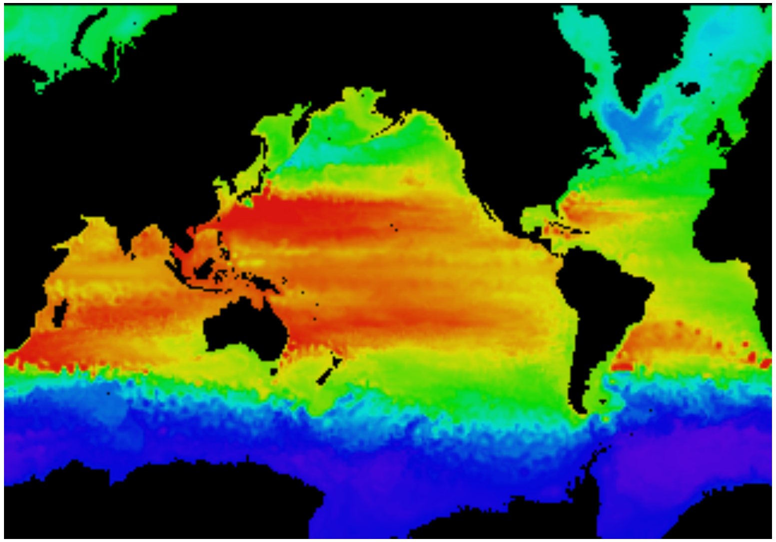
- **Problem is to compute:**

  f(latitude, longitude, elevation, time) → “weather” =

  (temperature, pressure, humidity, wind velocity)

- **Approach:**
  - *Discretize* the domain, e.g., a measurement point every 10 km
  - Devise an algorithm to predict weather at time t+δt given t

- **Uses:**
  - Predict major events, e.g., El Nino
  - Use in setting air emissions standards
  - Evaluate global warming scenarios

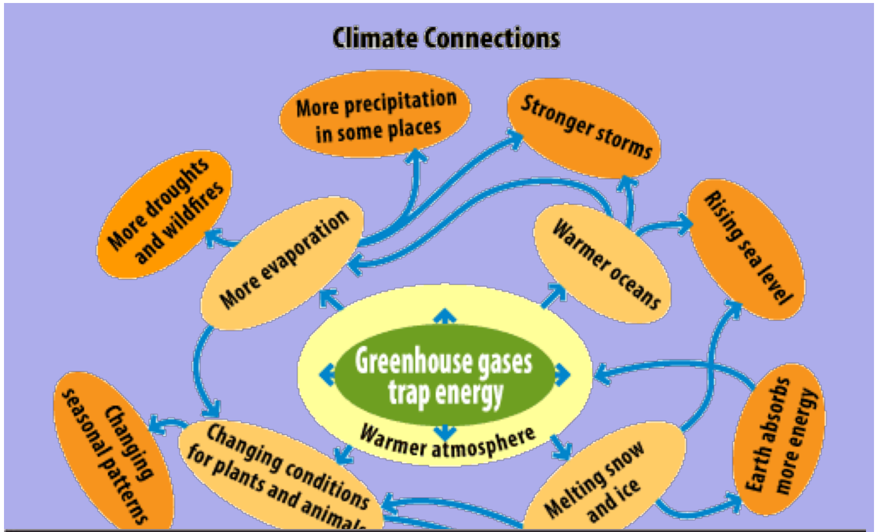


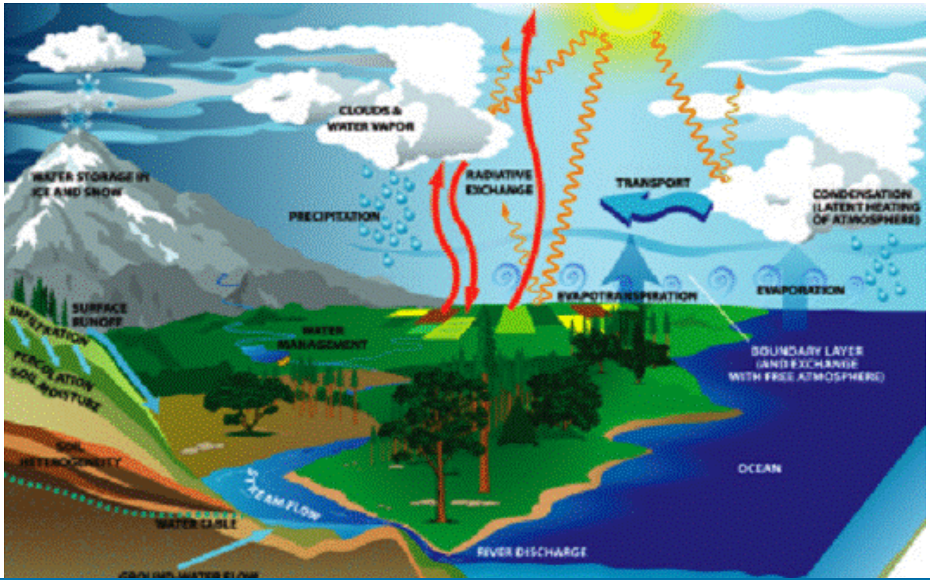
Source: http://www.epm.ornl.gov/chammp/chammp.html

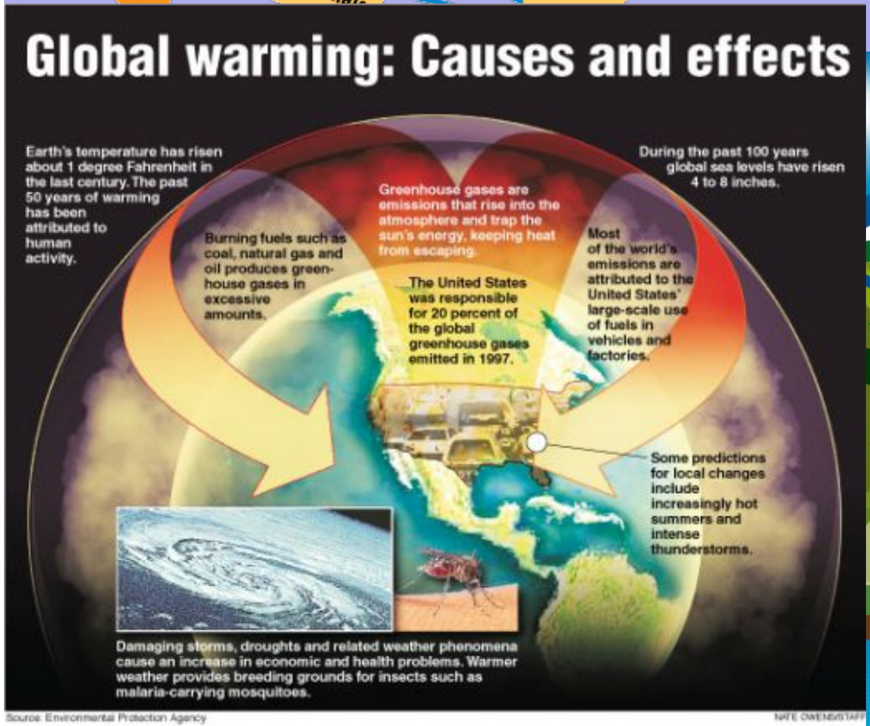
Global Climate Modeling Computation

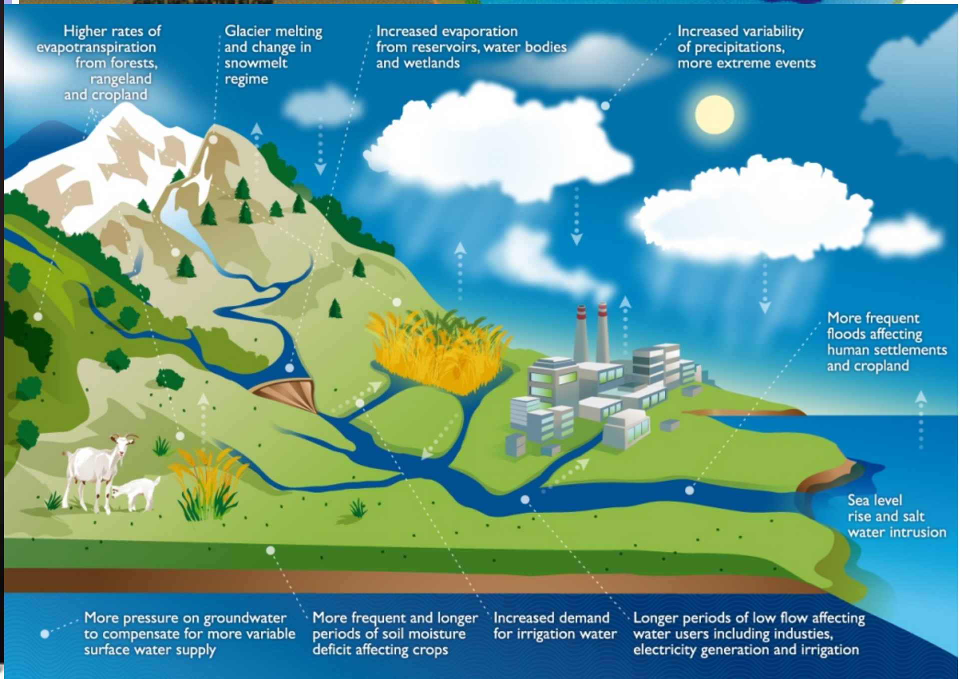
- One piece is modeling the fluid flow in the atmosphere
  - Solve Navier-Stokes equations
  - Roughly 100 Flops per grid point with 1 minute timestep
- Computational requirements:
  - Weather prediction (7 days in 24 hours) → 56 Gflop/s
  - Climate prediction (50 years in 30 days) →4.8 Tflop/s
  - To use in policy negotiations (50 years in 12 hours) → 288 Tflop/s
- To double the grid resolution, computation is 8x to 16x
- State of the art models require integration of atmosphere, clouds, ocean, sea-ice, land models, plus possibly carbon cycle, geochemistry and more
- Current models are coarser than this

超级计算与全球气候变化

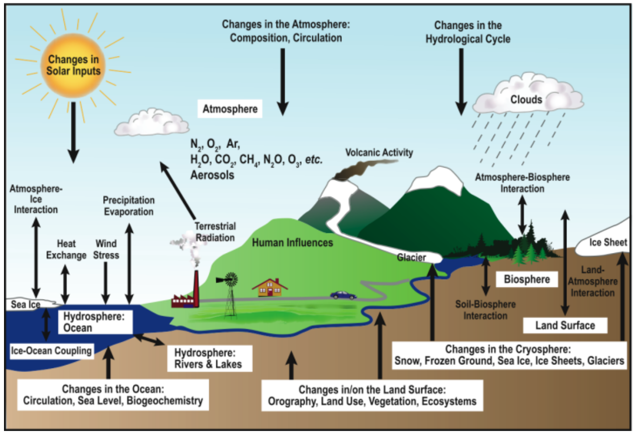








全球气候模拟



- 欧洲“Horizon 2020”研究与创新框架计划的E级超算计划将全球气候模拟作为重点应用领域之一；
- 日本的E级超算计划也将全球气候模拟作为重点应用领域之一；
- 美国DOE国家计算科学中心主任撰写的《Science Prospects And Benefits with Exascale Computing》报告，把全球气候模拟列为最有潜力的E级应用。
- 美国NCSA的Blue Water超算系统，也将全球气候模拟作为重点应用领域之一；

基于气候系统中的动力、物理、化学和生物过程建立数学方程组

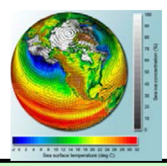
初始场

气候模式

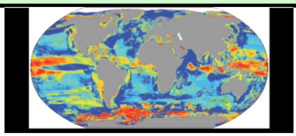
外强迫场

探索古气候变化

模拟当代气候

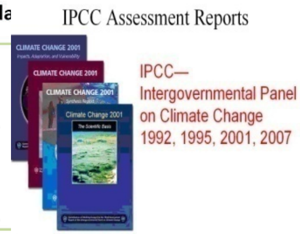


预测未来气候变化

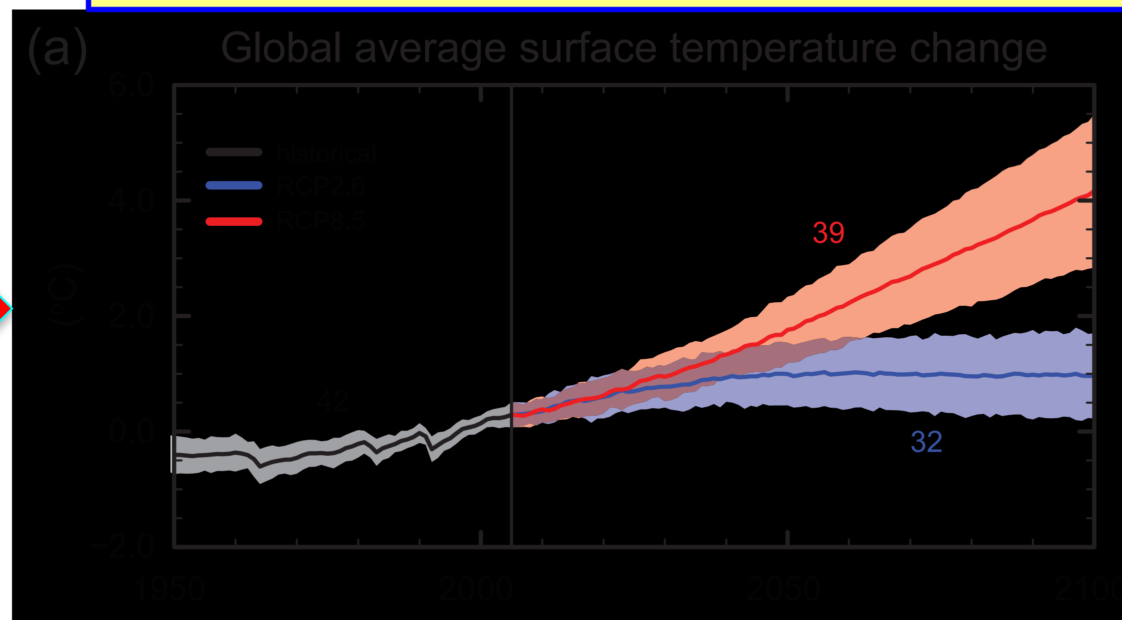


参与IPCC的模式

Model na	
ACCESS1.0, ACCESS1.3	Australia
BCC-CSM1.1, BCC-CSM1.1(m)	China
BNU-ESM	China
CanCM4	Canada
CanESM2	
CCSM4	USA
CESM1 (BGC)	
CESM1 (WACCM)	
CESM1 (FASTCHEM)	
CESM1 (CAM5)	
CESM1 (CAM5.1-FV2)	USA
CMCC-CM, CMCC-CMS	Italy
CMCC-CESM	
CNRM-CM5	France
CSIRO-Mk3.6.0	Australia
EC-EARTH	Europe
FGOALS-g2	China
FGOALS-s2	
FIO-ESM v1.0	China
GFDL-ESM2M, GFDL-ESM2G	USA
GFDL-CM2.1	
GFDL-CM3	
GISS-E2-R, GISS-E2-H	USA
GISS-E2-R-CC, GISS-E2-H-CC	
HadGEM2-ES	UK
HadGEM2-CC	
HadCM3	
HadGEM2-AO	Korea
INM-CM4	Russia
IPSL-CM5A-LR / -CM5A-MR / -CM5B-LR	France
MIROC4h, MIROC5	Japan
MIROC-ESM	
MIROC-ESM-CHEM	
MPI-ESM-LR / -ESM-MR / -ESM-P	Germany
MRI-ESM1	Japan
MRI-CGCM3	
NCEP-CFSv2	USA
NorESM1-M	Norway
NorESM1-ME	



IPCC（政府间气候变化专门委员会，2007年诺贝尔和平奖）给出的未来增温幅度完全是由参与IPCC评估的模式结果得出的。而全球气候变化谈判依据的正是这个结果。



只有努力发展我国自己的气候系统模式，提高模式模拟能力，才能增强我国在全球气候变化谈判中的话语权。

超级计算与核安全

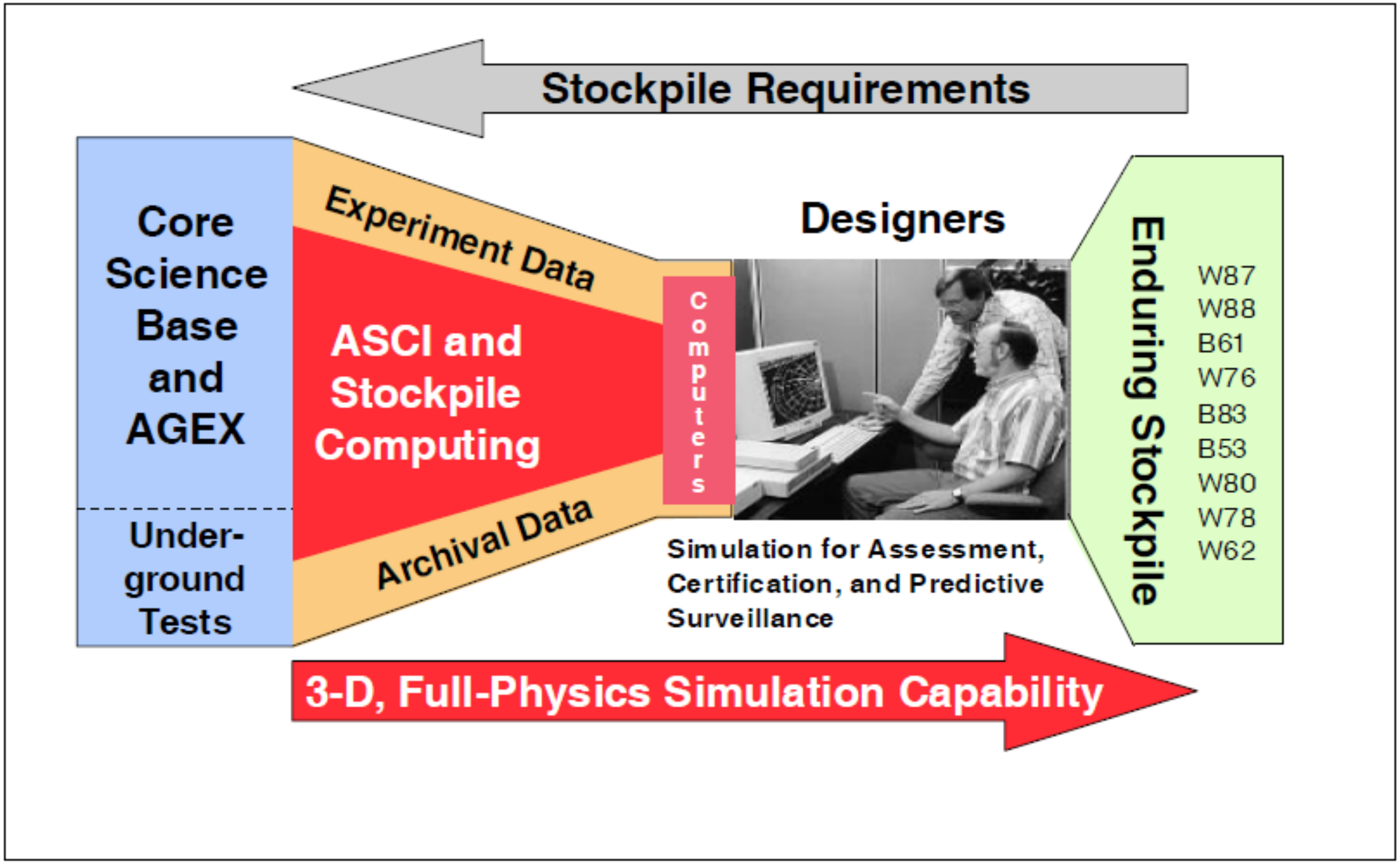
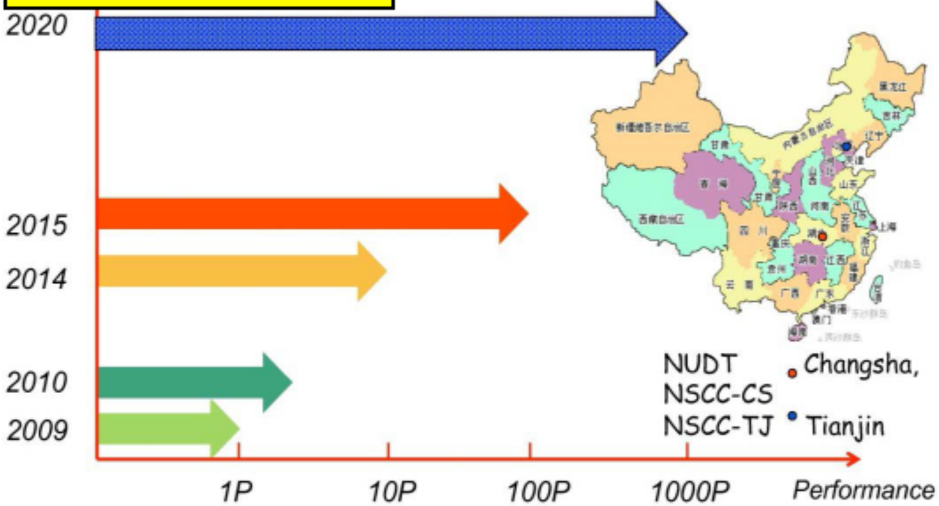


Figure 1. *Simulation, via ASCI, is the only practical, credible path from the known to the unknown for designers to provide an understanding of the enduring stockpile.*

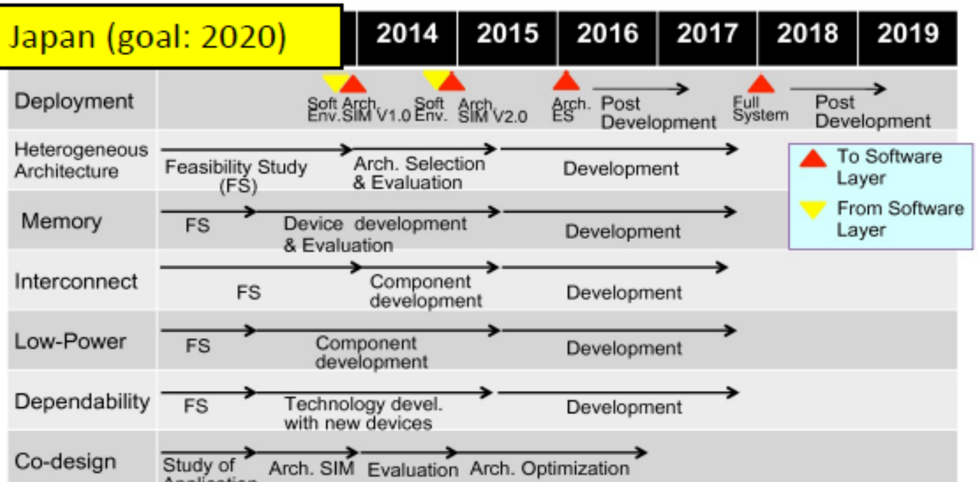
International Competitors Have Ambitious Plans

China (goal: 2020)



European Union (goal: 2020)

Japan (goal: 2020)

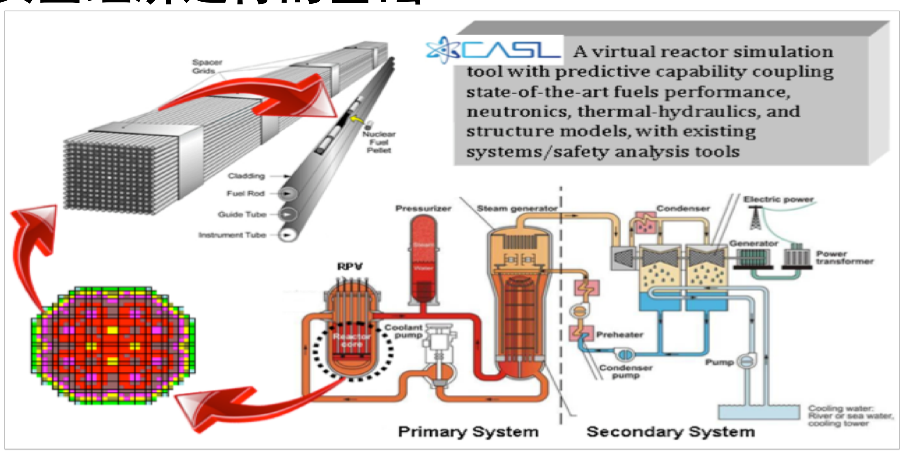


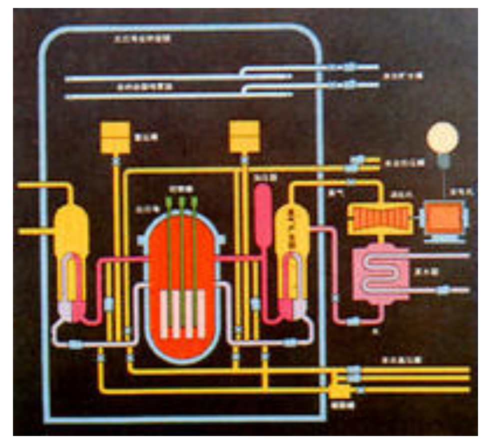
Other countries:
Russia
South Korea
India
Singapore
Australia

EU: J-Y. Berthou, IESP Workshop, Kobe, Japan, April 2012
China: Y. Lu, IESP Workshop, Kobe, Japan, April 2012
Japan: M. Kondo, IESP Workshop, Kobe, Japan, April 2012

核电站的安全性和经济性

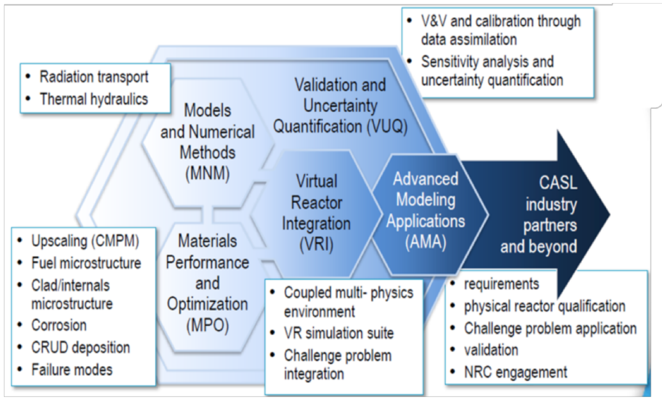
1. 世界上接近20%的电力来自核电。核电的安全性和经济性是政府、业主和公众重点关注的问题。
2. 在保证安全的前提下，实现高的组件卸料燃耗（长的燃料循环）、长的核电站使用寿命是核电追求的目标。
3. 核电站核岛（一回路）设备在高温、高压、强辐射、冷却剂腐蚀的苛刻环境中工作，设备在服役过程中的性能降级直接关系到核电站的安全性和经济性。
4. 核电站关键设备材料（组件、压力容器、堆内构件等）可靠性是保障核电站安全经济运行的基础。



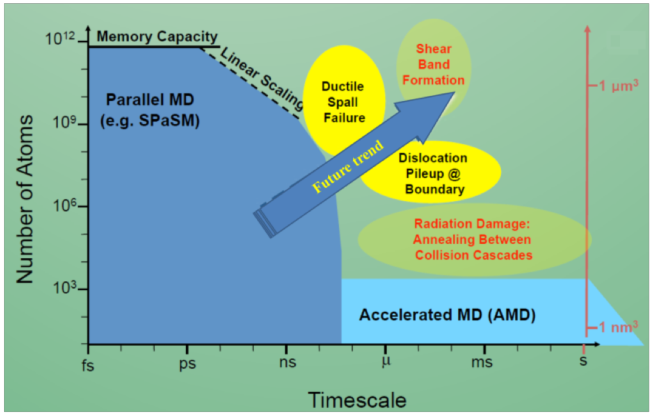


数值反应堆

（1）美国、欧洲等国家试图材料的基本理论出发，结合实验数据和运行经验，运用当代超强的计算机计算能力，通过模拟仿真，模拟核电站运行及部件的性能演化，进行验证和评估（V&V），预测核电站关键设备和材料的使用性能和寿命，提出数值反应堆概念（美国成为VERA，virtual environment for reactor application)。

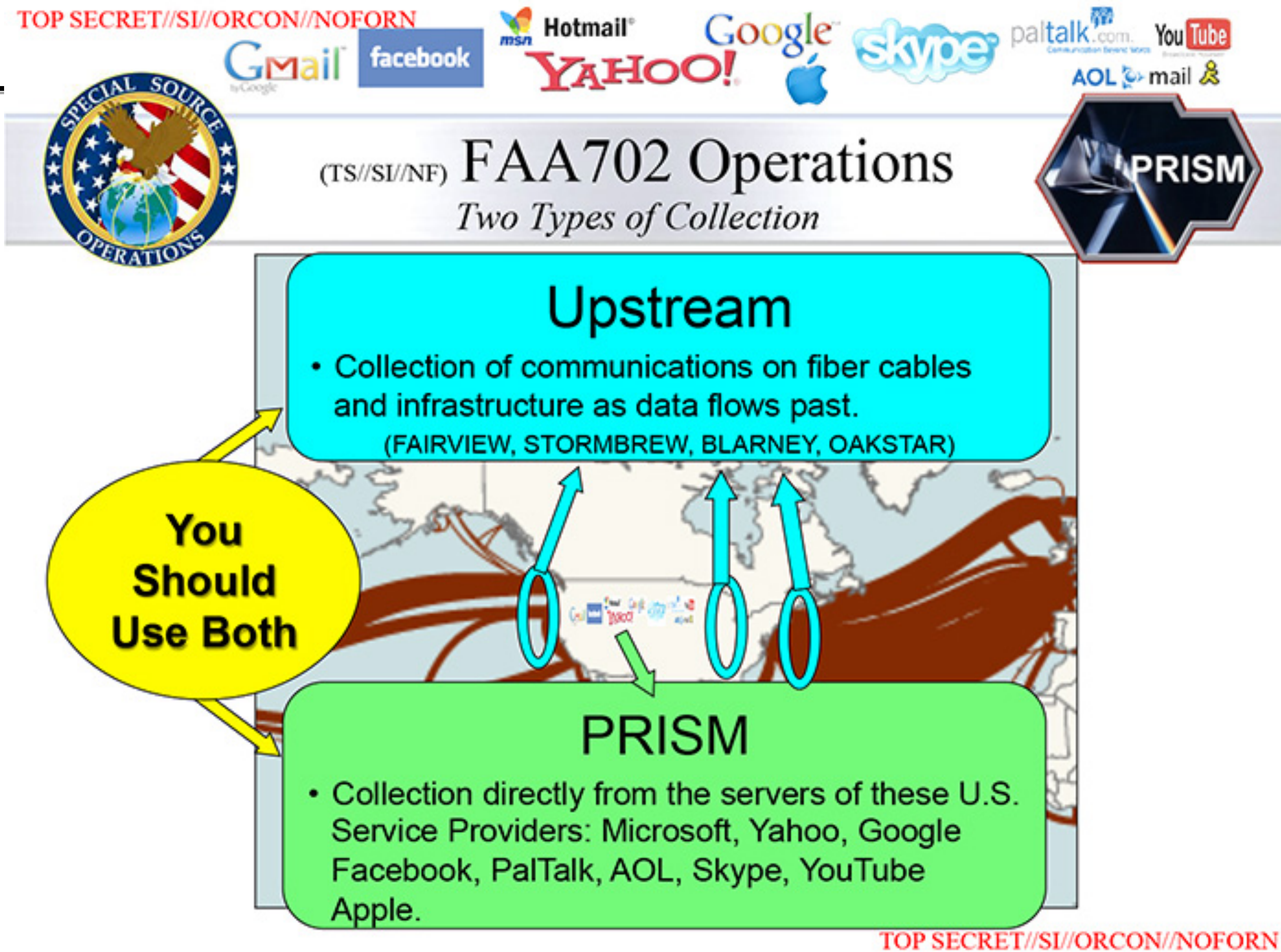


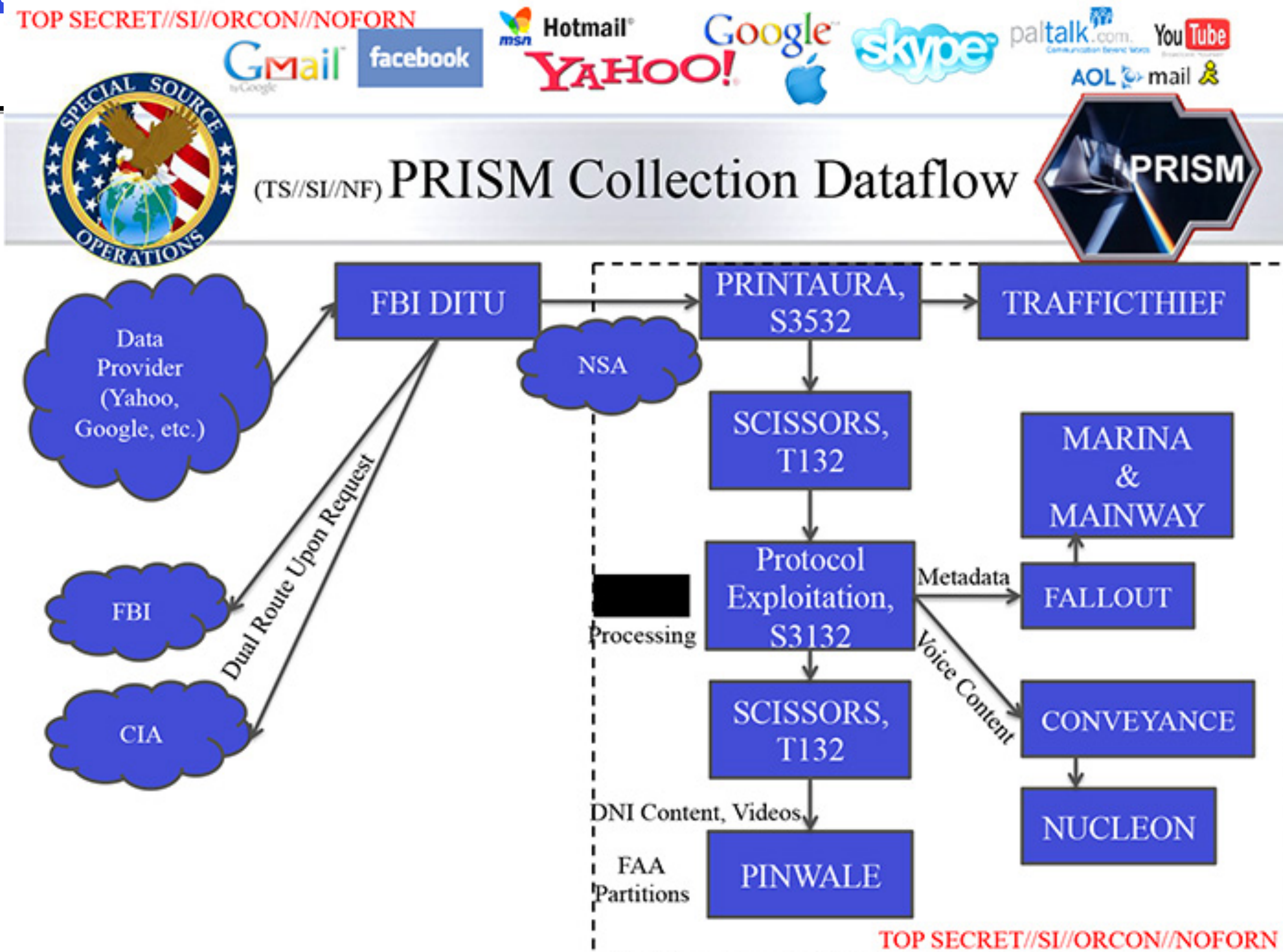
（2）美国的CASL计划，美欧的PVR计划等，重点关注燃料组件的CILC、GTRF、PCI和压力容器和堆内构件的辐照脆化等挑战性问题。目前这些计划已经取得阶段性成果。



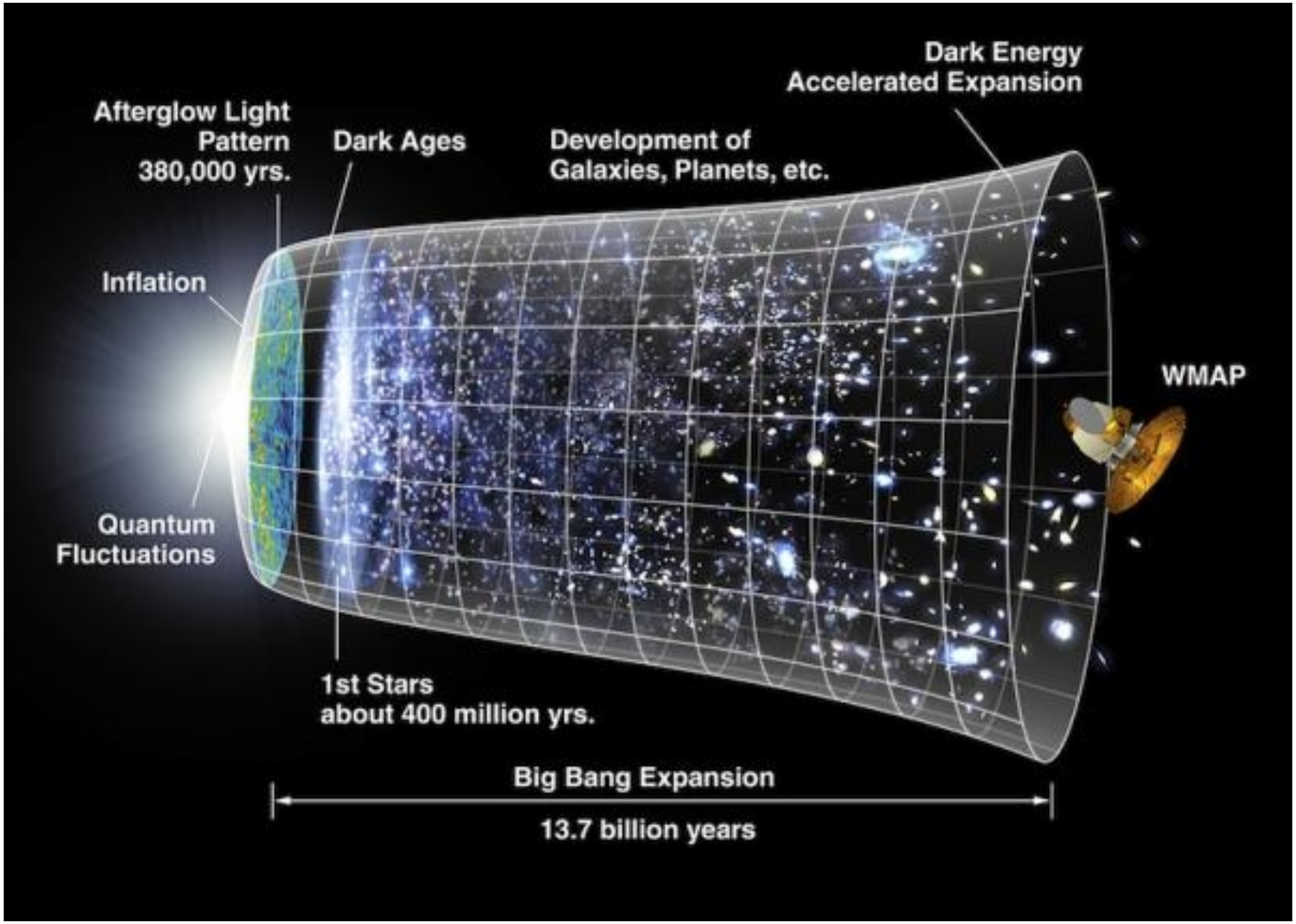
（3）CASL针对现有轻水堆延寿、升级等任务，利用超级计算机来研究轻水堆性能，开发反应堆分析数值模拟环境（VERA），通过高精度的模型来进行仿真模拟。

超级计算与国家信息安全


TOP SECRET//SI//ORCON//NOFORN
GMail
facebook
Hotmail
msn
YAHOO!
Google
skype
paltalk.com
YouTube
AOL mail
SPECIAL SOURCE OPERATIONS
(TS//SI//NF) FAA702 Operations
Two Types of Collection
PRISM
Upstream
• Collection of communications on fiber cables and infrastructure as data flows past.
(FAIRVIEW, STORMBREW, BLARNEY, OAKSTAR)
You Should Use Both
PRISM
• Collection directly from the servers of these U.S. Service Providers: Microsoft, Yahoo, Google Facebook, PalTalk, AOL, Skype, YouTube Apple.
TOP SECRET//SI//ORCON//NOFORN

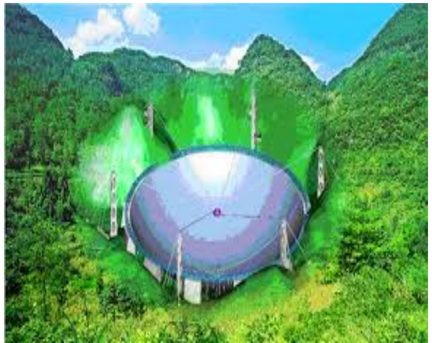

TOP SECRET//SI//ORCON//NOFORN
Gmail
facebook
msn Hotmail
Yahoo!
Google
skype
paltalk.com
YouTube
AOL mail
SPECIAL SOURCE OPERATIONS
(TS//SI//NF) PRISM Collection Dataflow
PRISM
Data Provider (Yahoo, Google, etc.)
FBI DITU
NSA
PRINTAURA, S3532
TRAFFICTHIEF
SCISSORS, T132
MARINA & MAINWAY
[redacted]
Processing
Protocol Exploitation, S3132
Metadata
FALLOUT
Voice Content
CONVEYANCE
NUCLEON
SCISSORS, T132
DNI Content, Videos
FAA Partitions
PINWALE
FBI
CIA
Dual Route Upon Request
TOP SECRET//SI//ORCON//NOFORN

超级计算与天文探索


Dark Energy
Accelerated Expansion
Afterglow Light
Pattern
380,000 yrs.
Dark Ages
Development of
Galaxies, Planets, etc.
Inflation
WMAP
Quantum
Fluctuations
1st Stars
about 400 million yrs.
Big Bang Expansion
13.7 billion years

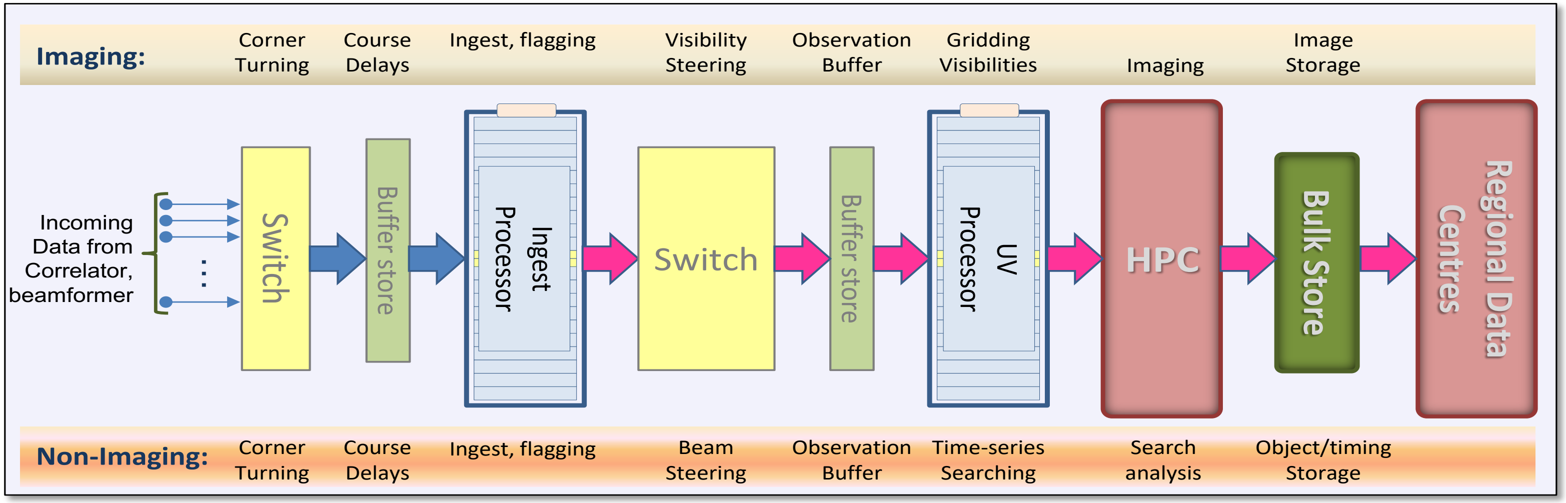
FAST大科学工程

- 500米口径球面射电望远镜FAST （Five hundred meters Aperture Spherical Telescope）是国家九大科技基础设施之一，建成后将成为世界上最大口径的射电望远镜，并将在20-30年内保持世界一流设备的地位。
- FAST落地贵州黔南州平塘县，已经于2013年12月顺利完成钢结构主动反射面环形支撑圈梁工程的合拢，一期将于2015-2016年初步建成。
- 围绕FAST的两个重要科学目标“脉冲星巡天”和“中性氢谱线巡天”，需要针对上百万个星系开展谱线处理、图像处理、噪声处理、干扰处理、宇宙参数模型拟合、脉冲星搜寻、高精度时间模型拟合等工作，迫切需要建设“实时监测+海量数据存储+高效精确高性能计算能力”的高性能计算平台。
- FAST一期，峰值数据将达到40TB/天（24小时*3600秒*1GB/2），峰值计算能力至少需要每秒浮点运算次数达200万亿次以上，存储至少在20PB以上。
- 随着FAST数据的长期积累，对存储和计算能力的需求还将进一步提升。



SKA大科学工程

- 平方公里阵列射电望远镜SKA（Square Kilometer Array）国际大科学工程是世界最大的射电望远镜项目，是我国作为正式成员参加的第二个国际大科学工程。
- SKA一期工程预计2020年完工，其存储和计算需求将是FAST的10到100倍。



SKA 时间线

2019	Operations SKA_1 2024: Operations SKA_2	
2019-2023	Construction of Full SKA, SKA_2	€1.5B
2016-2019	10% SKA construction, SKA_1	€300M
2012	Site selection	
2012 - 2016	Pre-Construction: 1 yr Detailed design PEP 3 yr Production Readiness	€90M
2008 - 2012	System design and refinement of specification	
2000 - 2007	Initial concepts stage	
1995 - 2000	Preliminary ideas and R&D	

SKA对超算的需求

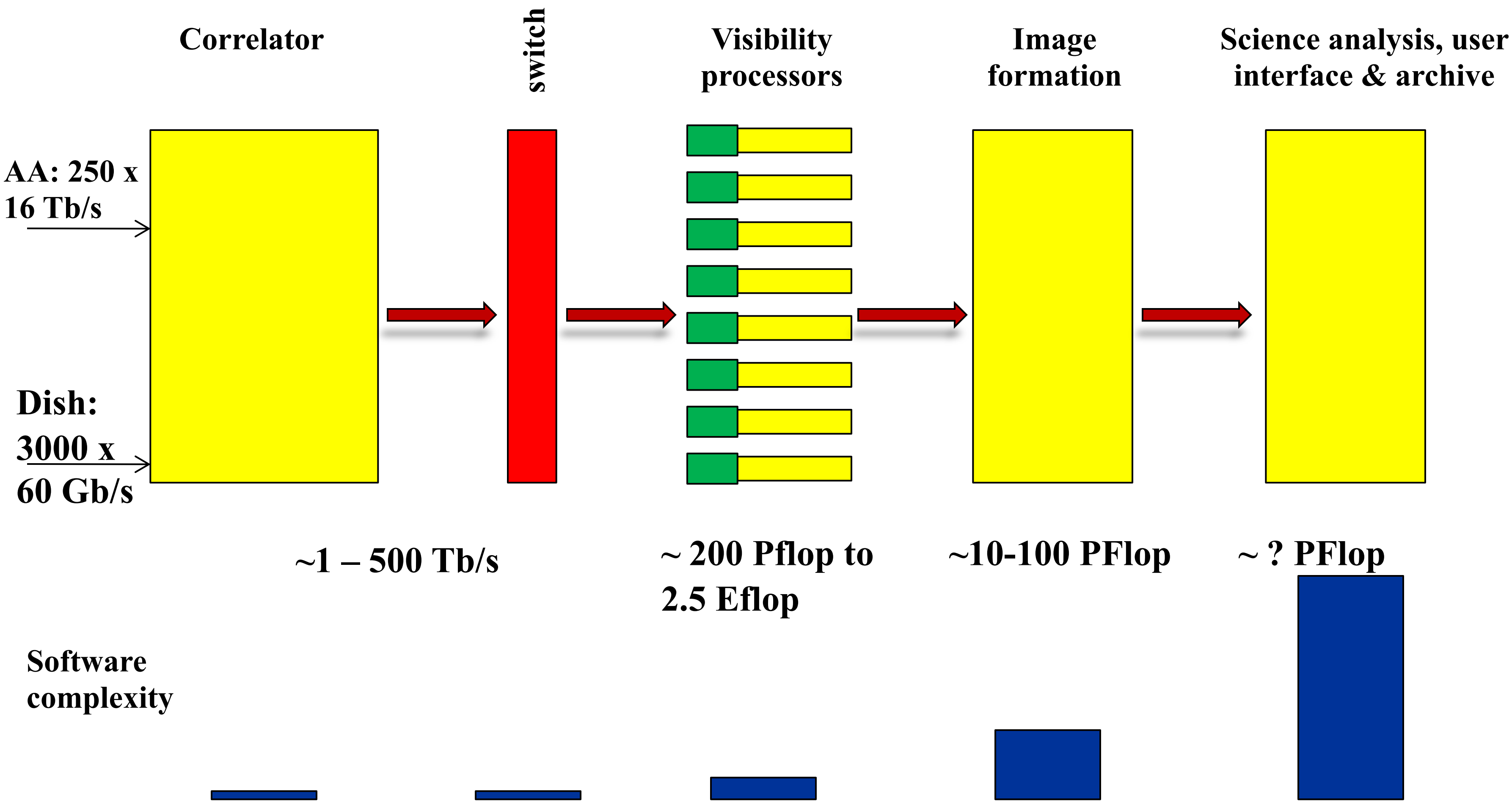
SKA1 LOW / SURVEY (36 beams):

- **Data rate out of correlator:**
  - 4670 GBytes/s (SURVEY),
  - 842 GBytes/s (LOW)
- **Max data rate into SDP: 995 GBytes/s**
  - (SURVEY: DRM Ch 3 H1 absorption, proportional to Nbeams, assuming 36)
- **Max computing load (flops/s): 32 Pflops**
  - (SURVEY: DRM Ch 3 H1 absorption, proportional to Nbeams, assuming 36)
- **Max UV buffer: 14 PBytes**
  - (SURVEY: DRM CH3 H1 absorption)

SKA1 Mid:

- **Data rate out of correlator:**
  - 1800 GBytes/s (BL design page 49)
- **Max data rate into SDP: 255 GBytes/s**
  - (DRM CH3: H1 absorption, band 1)
- **Max computing load: 10.0 Pflops/s**
  - (DRM CH3: H1 absorption, band 1)
- **Max UV buffer: 11.0 PBytes**
  - (DRM CH3: H1 absorption, band 1)

The SKA Processing Challenge



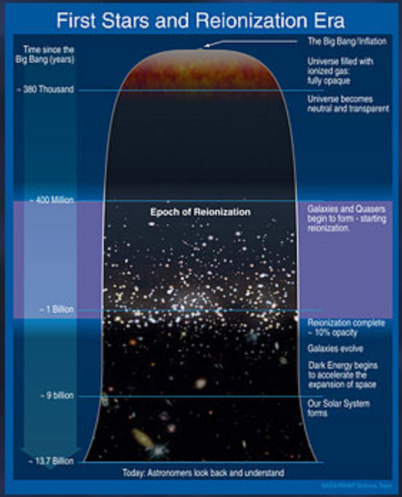
SKA Key Science Drivers

ORIGINS

➢ **Neutral hydrogen（中性氢） in the universe from the Epoch of Re-ionisation（再离子化） to now**

When did the first stars and galaxies form?
How did galaxies evolve?
Role of Active Galactic Nuclei活动星系核
Dark Energy, Dark Matter

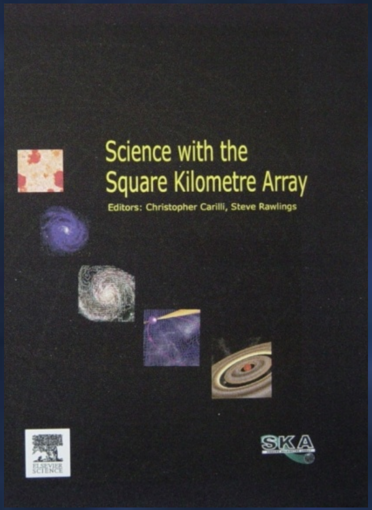
➢ **Cradle of Life（生命的摇篮）**



FUNDAMENTAL FORCES（基本力）

➢ **Pulsars, General Relativity & gravitational waves**

➢ **Origin & evolution of cosmic magnetism**

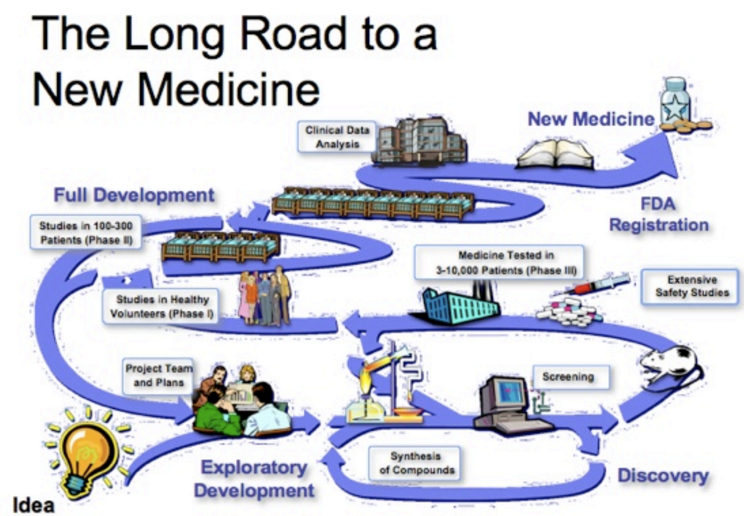


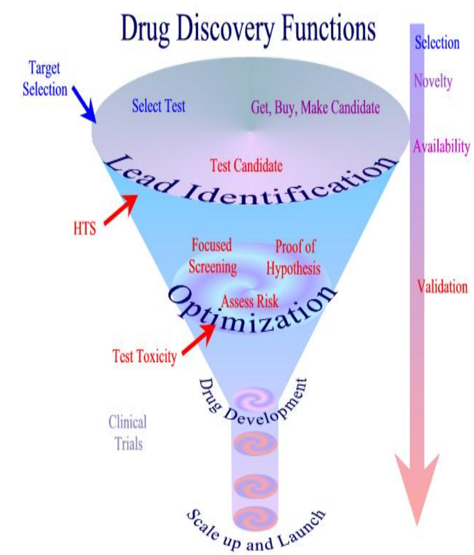
Science with the Square Kilometre Array (2004, eds. C. Carilli & S. Rawlings, *New Astron. Rev.*, 48)

TRANSIENTS (NEW PHENOMENA)瞬变事件

超级计算与新药研制

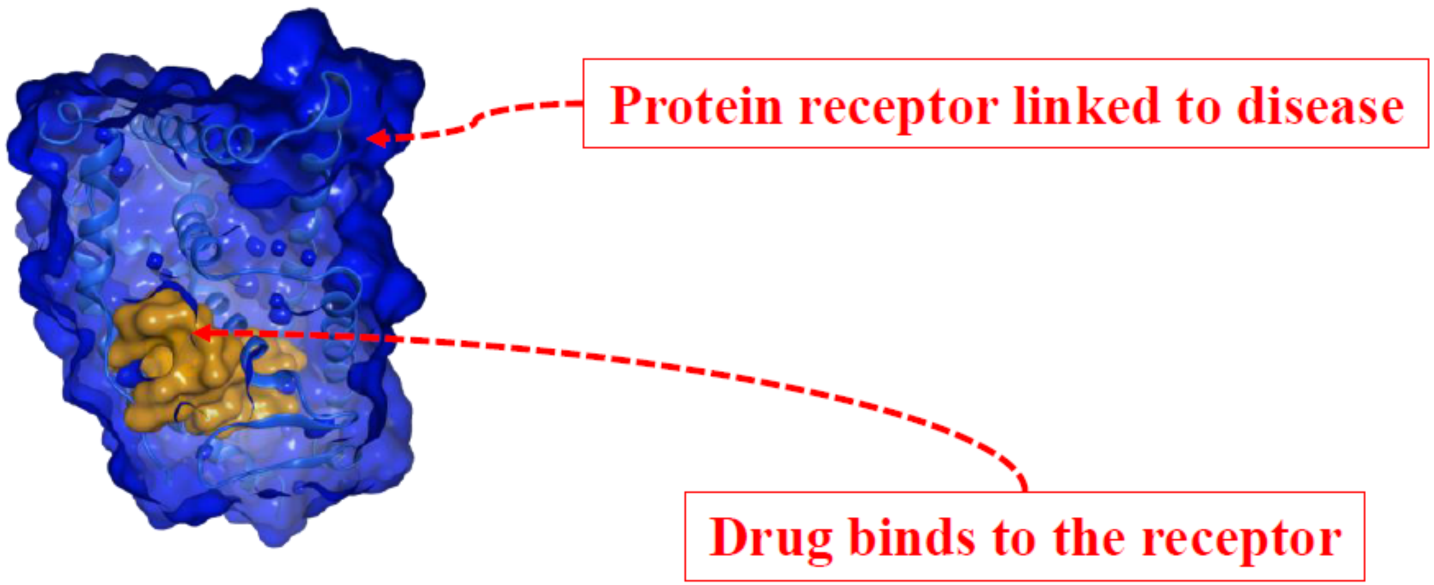
- 通常情况下，研制一款新药的投入从几亿到几十亿美元且需要10到15年的时间
- 工业界需要多快好省的从大量候选药物设计方案中尽快找出可行的方案
- 大部分的药物研制计划都以失败告终
- 每年全球制药企业的研发人员耗费精力研究数万种化合物，其中有些化合物对某种疾病尽管有特效，但由于其对应的人群太小，因而不具备商业开发的价值。这类药物被称为“坠落天使”(fallen angel)。





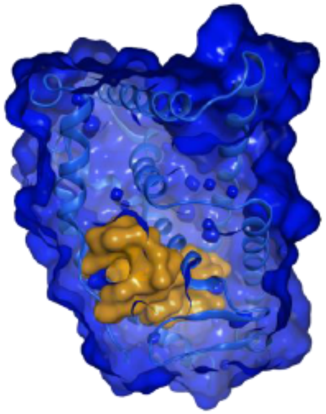
How do drugs do their ‘magic’?

Drug discovery/development paradigm (~2/3 of drugs) = a small organic molecule binding a protein receptor



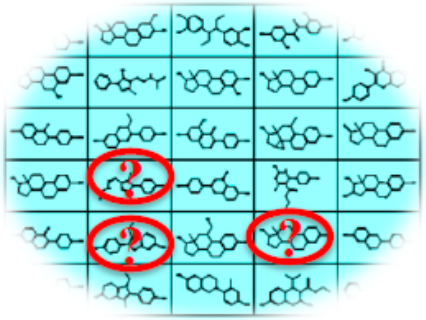
Drug discovery =
i) linking a disease with a protein and ii) finding small molecule effectors

Computational drug discovery & (virtual) screening



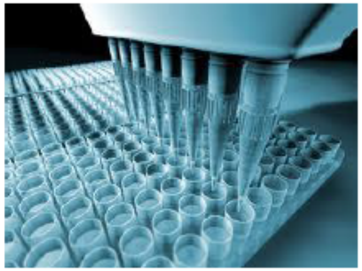
protein model

+



chemicals databases

Identify molecules that are likely to bind a given protein (*in vitro* binding assay)



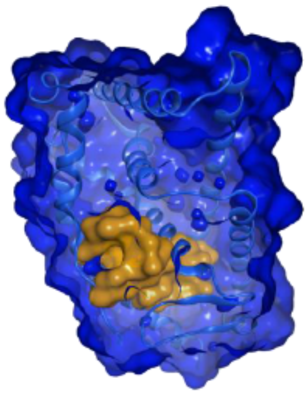
screening 1 drug candidate: \$10 to \$20

Chemical databases out there : up to 80M molecules

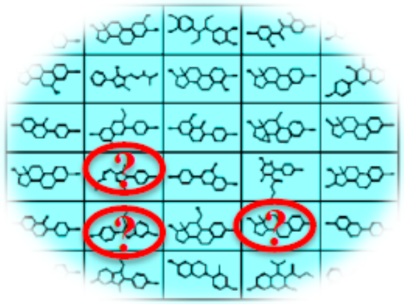
Virtual screening :
Virtual *in vitro* binding assays

Virtual docking of chemicals in proteins:

identify molecules that may bind in a protein

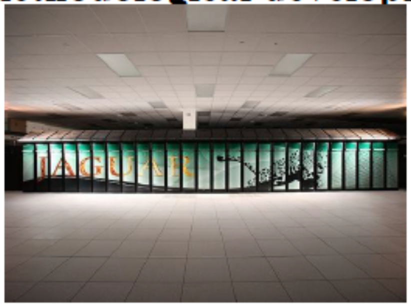


+



Important technology in pharmaceutical industry: drug discovery (hits) and drug design (lead optimization)

methodological developments



Leverage the power of the world's top supercomputers

<u>An interdisciplinary approach:</u>

Computational Sciences

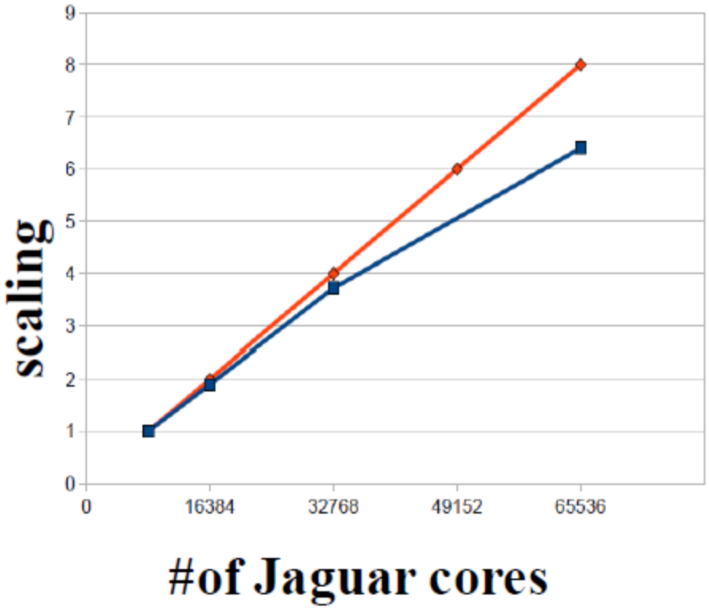
Structural Biology, Biophysics, Biochemistry

Chemistry/Medicinal Chemistry

Parallelization of Autodock on Jaguar

(Collignon, Schulz, Smith, Baudry. *J. Comp. Chem*. 2011,)

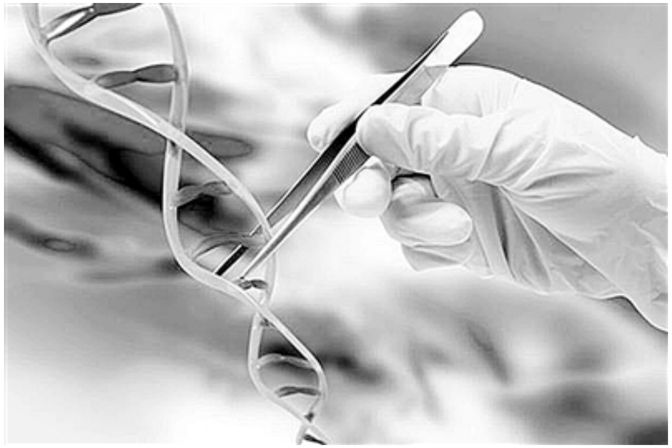
Our Autodock4 MPI parallelization



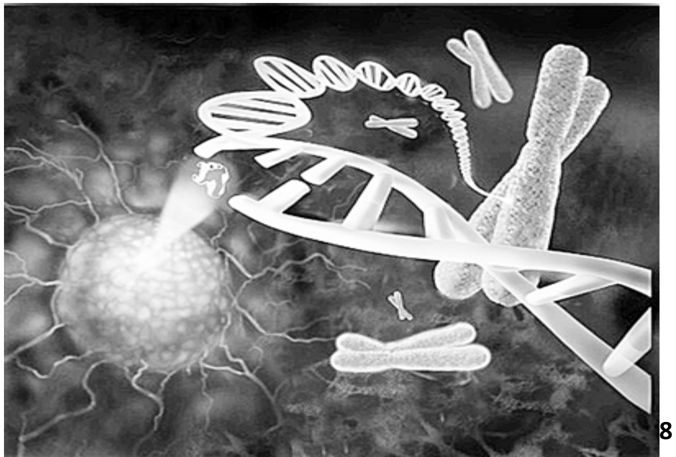
before data optimization:	1 protein	10^4 ligands	1 day
Presently:	1 protein	10^6 ligands	1 day
	10 proteins	10^5 ligands	1 day
Exascale (2018):	100 proteins	10^7 ligands	1 day

超级计算与个性化医疗

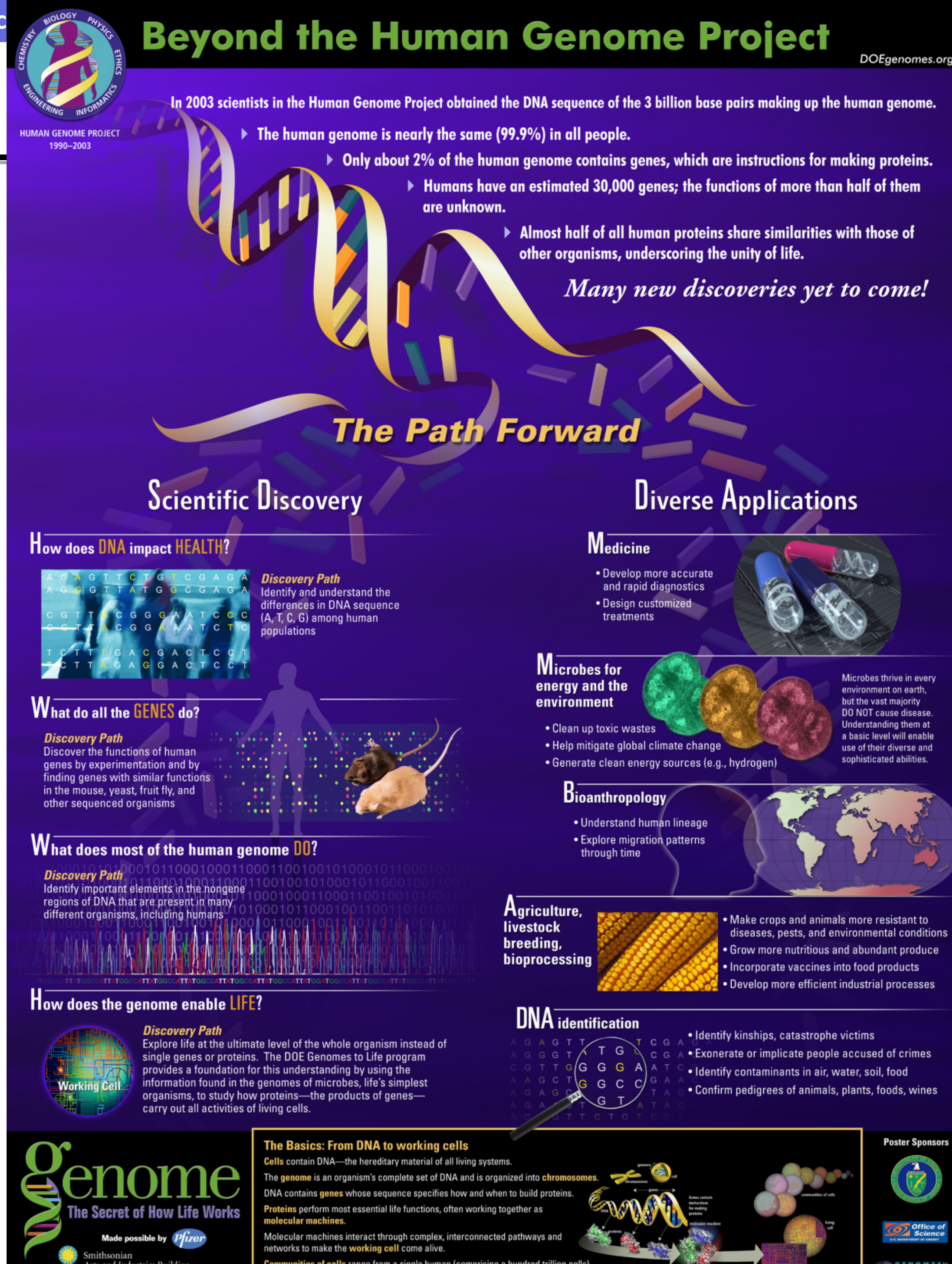
- 2013年5月，好莱坞影星安吉丽娜·茱莉做出一项轰动全球的决定：切除双侧乳腺。她的乳腺并未病变，但仍然执意手术，支持她的是一次价值4000美元的基因检测：检测显示，她体内携带有BRCA1型基因突变。
- 小小基因的个体差异巨大，会影响药物治疗效果
  - 一个令人惊愕的数据：在所有被服用抗癌药物中，只有25%起作用，其它75%几乎被浪费。
  - 药物对不同个体的疗效差异达300倍，一个不起眼的基因突变往往会使具有高度靶向性的抗癌药物完全无效
- 对所有病人肿瘤样本基因组进行完整测序，把肿瘤细胞基因序列和病人正常细胞基因序列进行对比，验证病人的癌症基因变异



- 基因组分析能在找出肿瘤基因组中的"驱动"变异，并确定以前携带有这些变异的细胞经过治疗有没有被清除。
- 利用“深度数字测序”技术，对样本每个变异进行上千次测序，获得变异频率，绘制癌细胞基因随病情发展的演化过程。随着癌症发展，肿瘤需要新的变异，但仍要保持原来在最初位置导致癌变的那部分变异群。



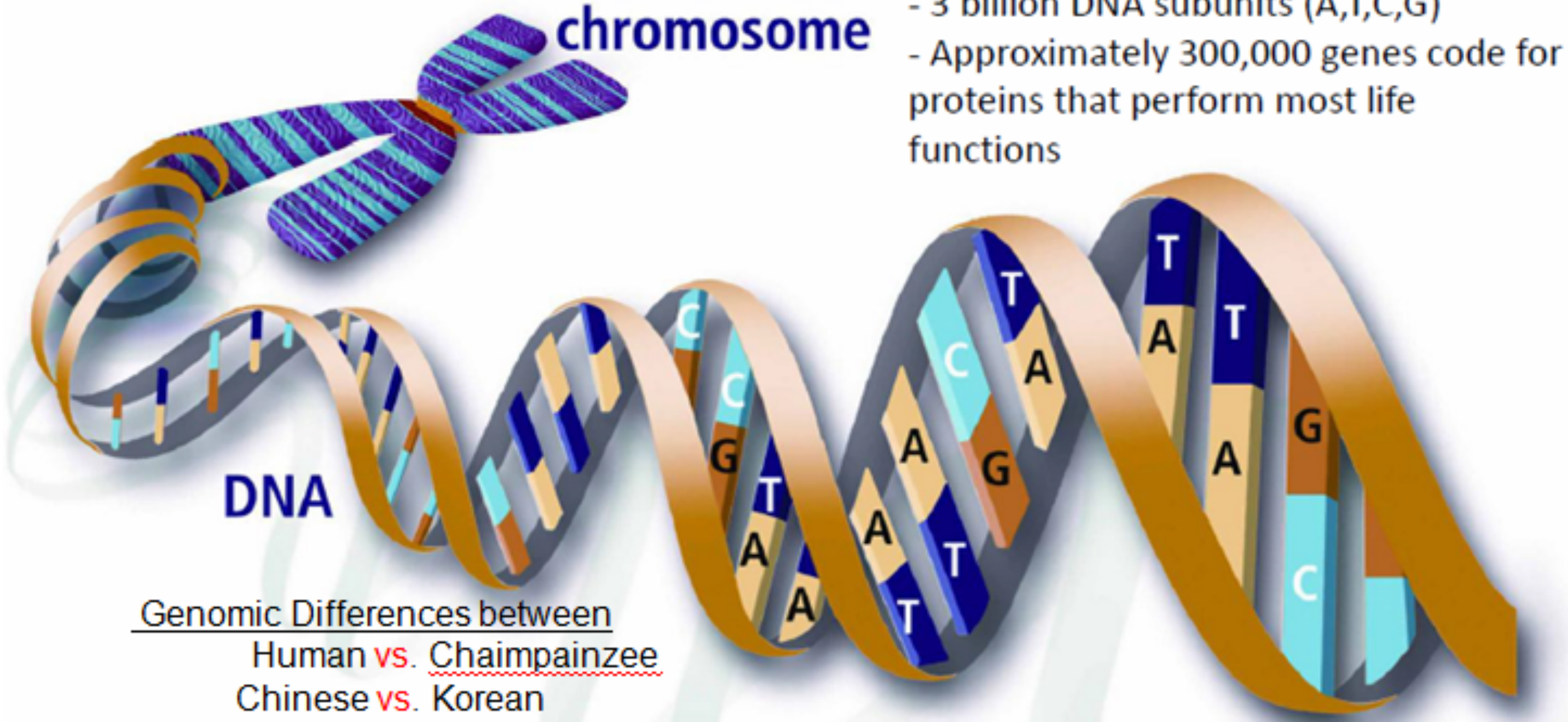
- 如果直接瞄准早期基因变异，标靶药物效果会更好。
- 如果瞄准后期才出现的癌细胞专有变异，可能就没什么效果，不能杀死所有的肿瘤细胞。
- 通过基因测序，医生能预先知道某种药物会对哪些人管用。


Beyond the Human Genome Project
DOEgenomes.org
BIOLOGY PHYSICS ETHICS INFORMATICS ENGINEERING CHEMISTRY
HUMAN GENOME PROJECT 1990–2003
In 2003 scientists in the Human Genome Project obtained the DNA sequence of the 3 billion base pairs making up the human genome.
The human genome is nearly the same (99.9%) in all people.
Only about 2% of the human genome contains genes, which are instructions for making proteins.
Humans have an estimated 30,000 genes; the functions of more than half of them are unknown.
Almost half of all human proteins share similarities with those of other organisms, underscoring the unity of life.
Many new discoveries yet to come!
The Path Forward
Scientific Discovery
Diverse Applications
How does DNA impact HEALTH?
Discovery Path
Identify and understand the differences in DNA sequence (A, T, C, G) among human populations
What do all the GENES do?
Discovery Path
Discover the functions of human genes by experimentation and by finding genes with similar functions in the mouse, yeast, fruit fly, and other sequenced organisms
What does most of the human genome DO?
Discovery Path
Identify important elements in the nongene regions of DNA that are present in many different organisms, including humans
How does the genome enable LIFE?
Discovery Path
Explore life at the ultimate level of the whole organism instead of single genes or proteins. The DOE Genomes to Life program provides a foundation for this understanding by using the information found in the genomes of microbes, life's simplest organisms, to study how proteins—the products of genes—carry out all activities of living cells.
Working Cell
Medicine
• Develop more accurate and rapid diagnostics
• Design customized treatments
Microbes for energy and the environment
• Clean up toxic wastes
• Help mitigate global climate change
• Generate clean energy sources (e.g., hydrogen)
Microbes thrive in every environment on earth, but the vast majority DO NOT cause disease. Understanding them at a basic level will enable use of their diverse and sophisticated abilities.
Bioanthropology
• Understand human lineage
• Explore migration patterns through time
Agriculture, livestock breeding, bioprocessing
• Make crops and animals more resistant to diseases, pests, and environmental conditions
• Grow more nutritious and abundant produce
• Incorporate vaccines into food products
• Develop more efficient industrial processes
DNA identification
• Identify kinships, catastrophe victims
• Exonerate or implicate people accused of crimes
• Identify contaminants in air, water, soil, food
• Confirm pedigrees of animals, plants, foods, wines
genome
The Secret of How Life Works
Made possible by Pfizer
Smithsonian
The Basics: From DNA to working cells
Cells contain DNA—the hereditary material of all living systems.
The genome is an organism's complete set of DNA and is organized into chromosomes.
DNA contains genes whose sequence specifies how and when to build proteins.
Proteins perform most essential life functions, often working together as molecular machines.
Molecular machines interact through complex, interconnected pathways and networks to make the working cell come alive.
Poster Sponsors
Office of Science

Genomic sequencing and personalized medicine
基因排序&私人定制医疗

Human genome reference database: 3 GB
Working data set: 40-100 X 3 GB = 120-300 GB

Facts: Human Genome
- Trillions of cells
- 23 pairs of chromosomes
- 2 meters of DNA
- 3 billion DNA subunits (A,T,C,G)
- Approximately 300,000 genes code for proteins that perform most life functions



Genomic Differences between
Human vs. Chaimpainzee
Chinese vs. Korean
David Chen vs. President Hu
Myself vs. my son
When I am healthy vs. when I have a cancer

From Wang Bingqian's presentation at NVIDIA User Conference, Beijing, Dec 2011

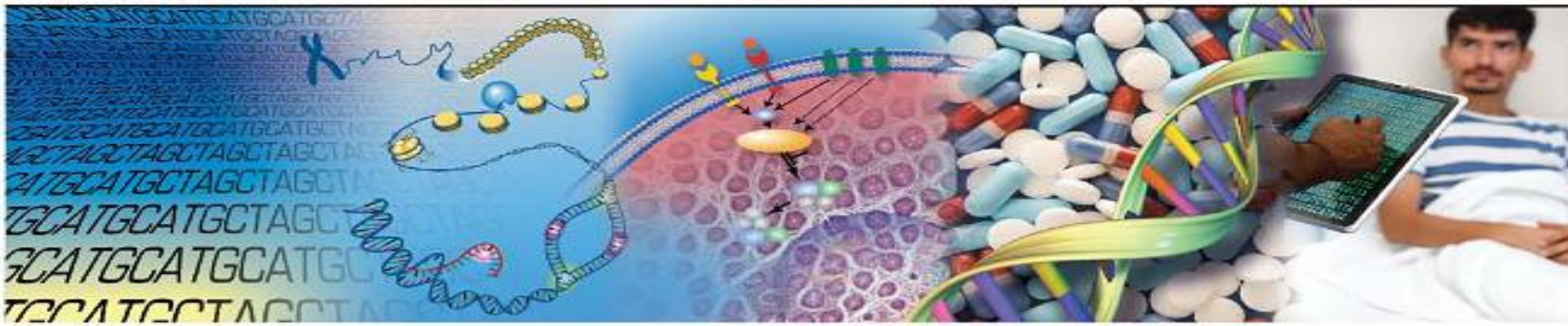
Personalized medicine saved Dr. Lukas Wartman

- 2002: Lukas Wartman about to graduate from WashU Medical School
- 2003: Dr. Lukas Wartman, 25, diagnosed with Leukemia
- 2004: chemotherapy, cancer in remission
- 2011: cancer relapsed, marrow transplant didn't work
- 2011: WashU doctors started genomic medicine effort
- Aug31: WashU Genome Center started sequencing Dr. Wartman's cancer DNA and RNA
- Sept: RNA sequencing revealed overactive FLT3 gene in cancer cell
- October: Sutent was administrated and Leukemia in full remission

http://www.nytimes.com/2012/07/08/health/

私人定制的治疗计划

Understanding the Structure of Genomes | Understanding the Biology of Genomes | Understanding the Biology of Disease | Advancing the Science of Medicine | Improving the Effectiveness of Healthcare



Sequencing data collection

focus on very large data generation, mainly from 1000 whole genome sequencing, and the data processing and reduction

includes plant, animal, and microbiome genomics

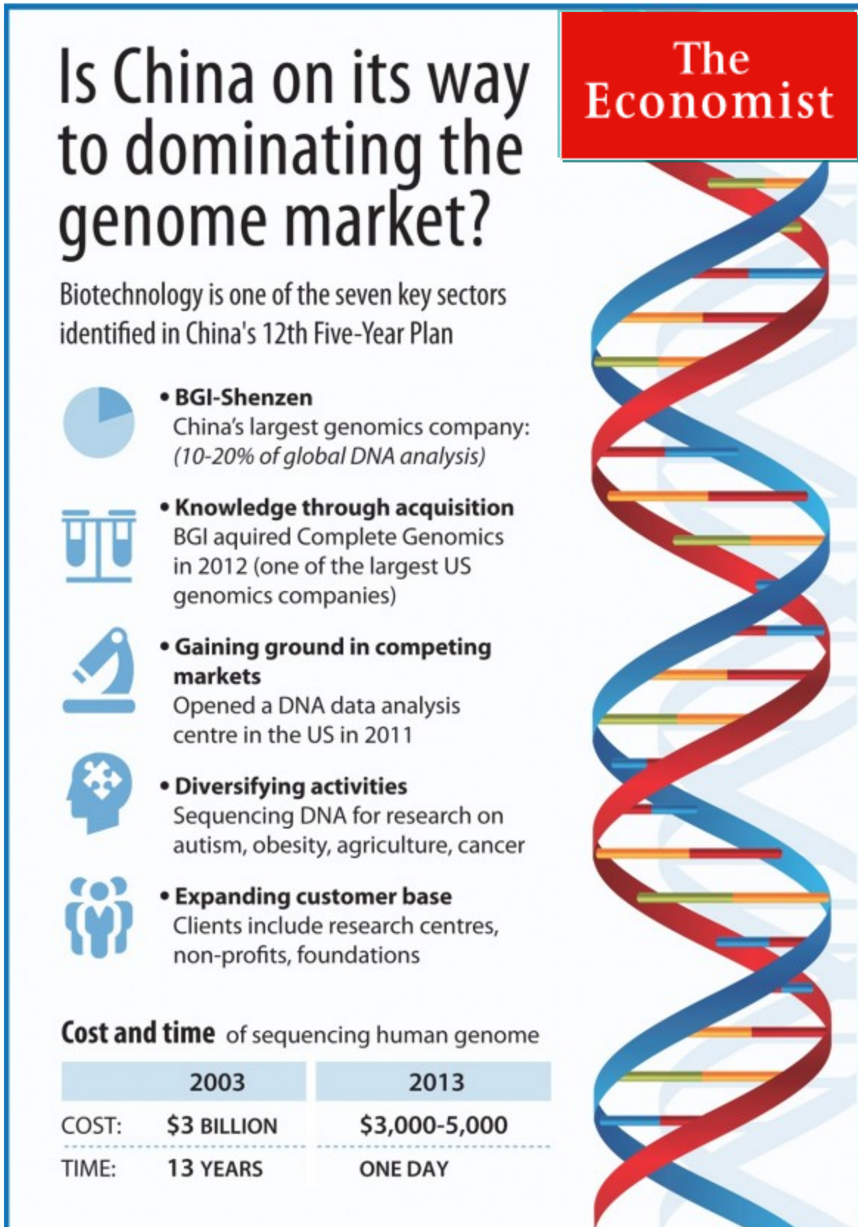
Translational medicine

focus on data integration including genomic data, and the analytics required to identify biomarkers, understand disease mechanisms, and to identify new medical treatments

Personalized healthcare

focus on delivering genomic medicine to improve Outcomes

NHGRI, a branch of NIH, has defined 5 steps for genomic medicine. *(E. Green et al., Nature 470)*



- 中国首套Illumina HiSeq X10测序系统已经于2014年5月15日运抵天津武清区诺禾致源公司创业总部基地的实验室.公司总裁李瑞强

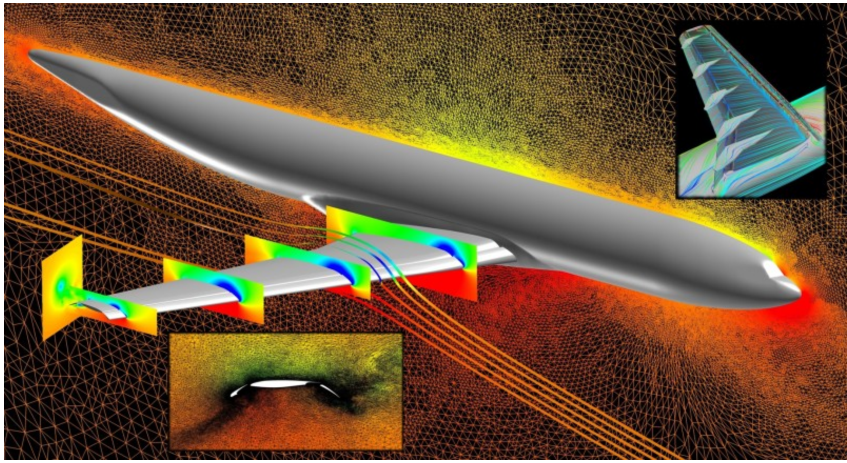
- 药明康德的第二套X10也将于7月开始运行

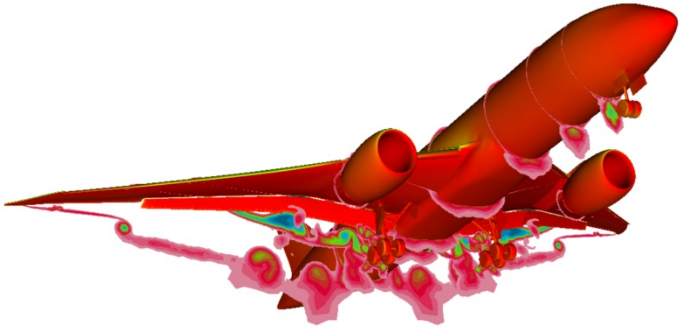
- 深圳华大基因主要的测序平台为128台HiSeq 2000. 华大的大数据战略依赖的测序平台是2013年收购的美国测序仪公司Complete Genomics 的测序仪（BGISEQ1000）

http://m.china.caixin.com/m/2014-05-19/100679036.html

超级计算与航空航天

- NASA aeronautical innovators are helping to design future airliners that will cut fuel consumption, reduce polluting emissions and fly more quietly.
- Future aircraft designs routinely flying during the 2030's may look very different from today's airliners in order to deliver on the promises of reduced fuel burn, noise and emissions.
- Yet in computational fluid dynamics, or CFD, the design tools that helped give us the modern airliners flying today are not expected to be up to the challenge in the future without some serious upgrades.
- In the meantime, even as NASA's CFD experts work down a path toward their long-range future goals of 2030 --





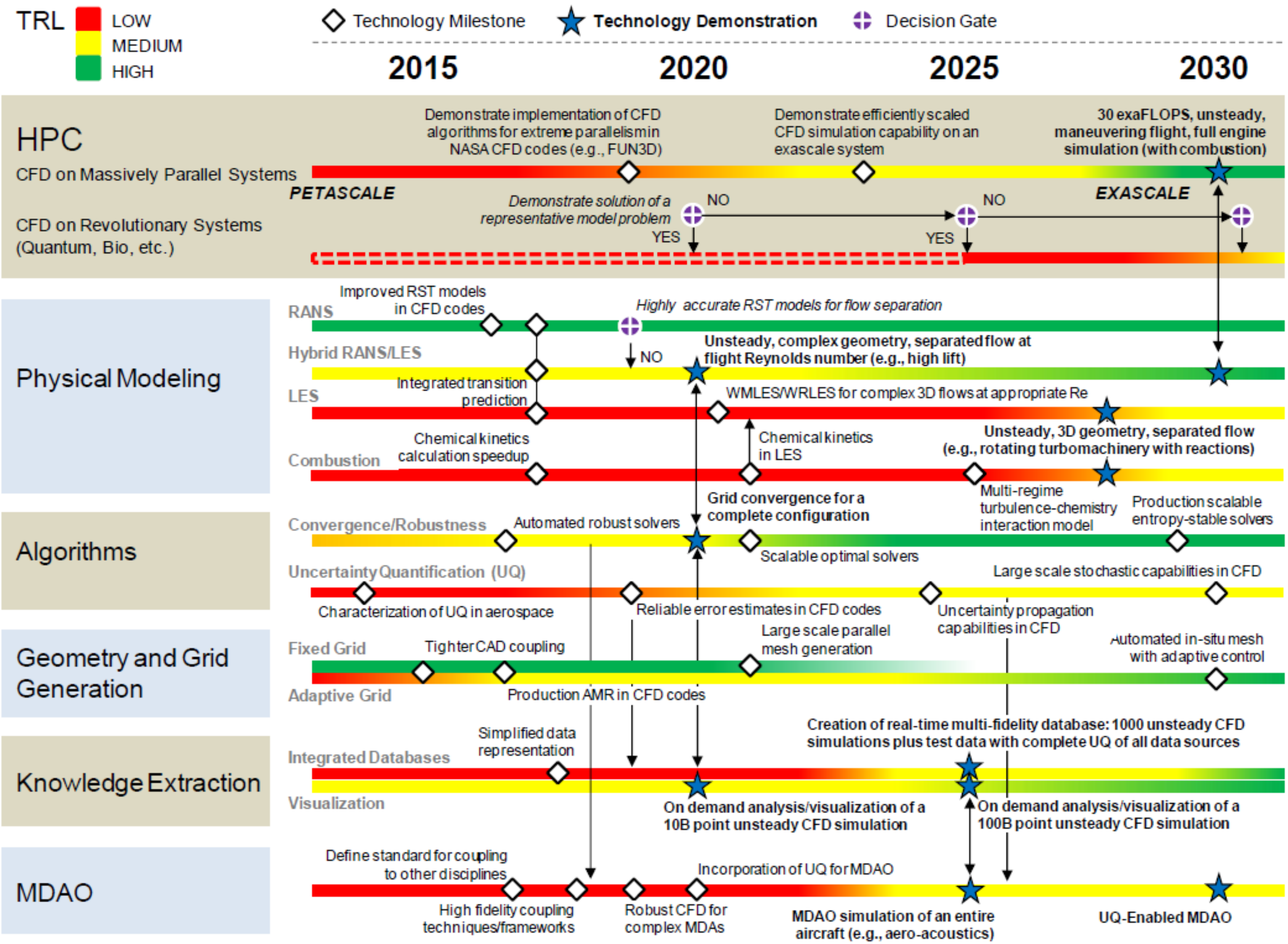


Figure 1. Technology Development Roadmap

HPC Roadmap

Hardware centric → data & process Centric
HPC的发展史：从硬件主机为中心 → 数据和流程为中心

Data and Process (work flow) Centric

Software Centric

Hardware System Centric

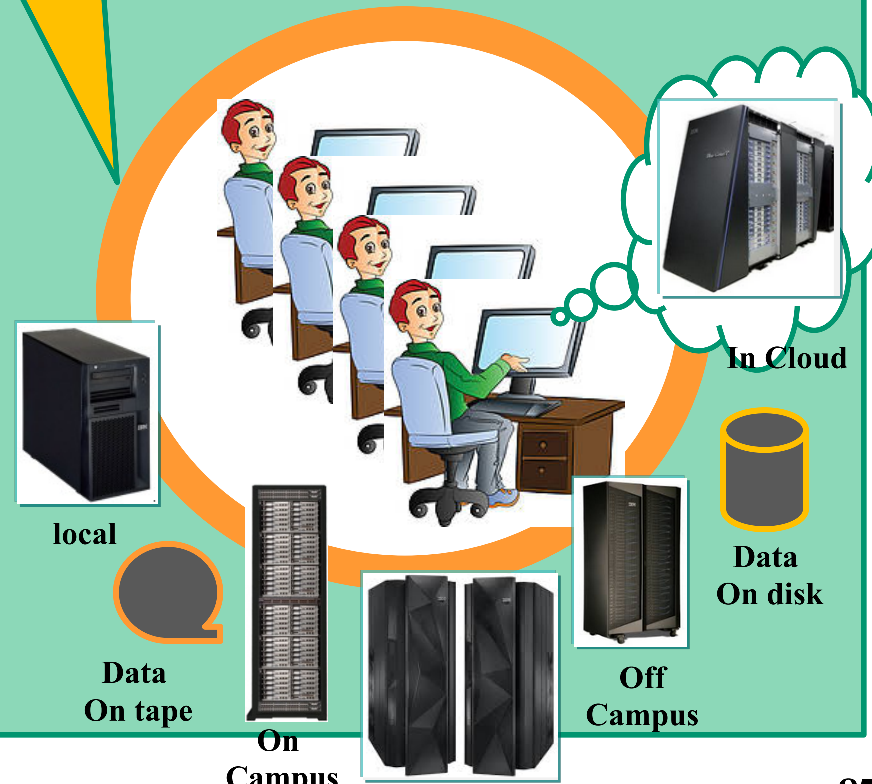
In the Past – Hardware Centric

Limited HPC system to use
Develop your own app
Cut your job size to fit h/w
Need strong computer skills

Enabling Software

Today and Future – User Centric

Lots of HPC systems to use
Open source or 3rd party app
Most jobs well fit with h/w
No need for much computer skills



Outline

- Why powerful computers must be parallel processors
  - Including your laptops and handhelds
- Large CSE problems require powerful computers
  - Commercial problems too
- Why writing (fast) parallel programs is hard
  - But things are improving

Principles of Parallel Computing

- Finding enough parallelism (Amdahl's Law)
- Granularity – how big should each parallel task be
- Locality – moving data costs more than arithmetic
- Load balance – don't want 1K processors to wait for one slow one
- Coordination and synchronization – sharing data safely
- Performance modeling/debugging/tuning

➡ All of these things makes parallel programming even harder than sequential programming.

"Automatic" Parallelism in Modern Machines

- **Bit level parallelism**
  - within floating point operations, etc.
- **Instruction level parallelism (ILP)**
  - multiple instructions execute per clock cycle
- **Memory system parallelism**
  - overlap of memory operations with computation
- **OS parallelism**
  - multiple jobs run in parallel on commodity SMPs

Limits to all of these -- for very high performance, need user to identify, schedule and coordinate parallel tasks

Finding Enough Parallelism

- Suppose only part of an application seems parallel
- Amdahl's law
  - let s be the fraction of work done sequentially, so (1-s) is fraction parallelizable
  - P = number of processors

$$\text{Speedup}(P) = \text{Time}(1)/\text{Time}(P)$$

$$<= 1/(s + (1-s)/P)$$

$$<= 1/s$$

- Even if the parallel part speeds up perfectly performance is limited by the sequential part
- Top500 list: currently fastest machine has P~3.1M; 2nd fastest has ~560K

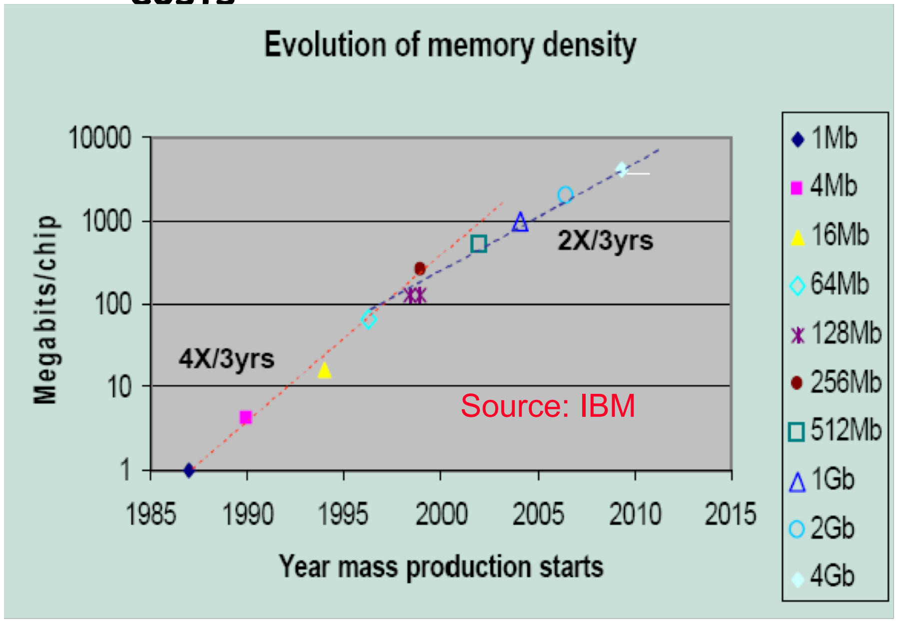
Overhead of Parallelism

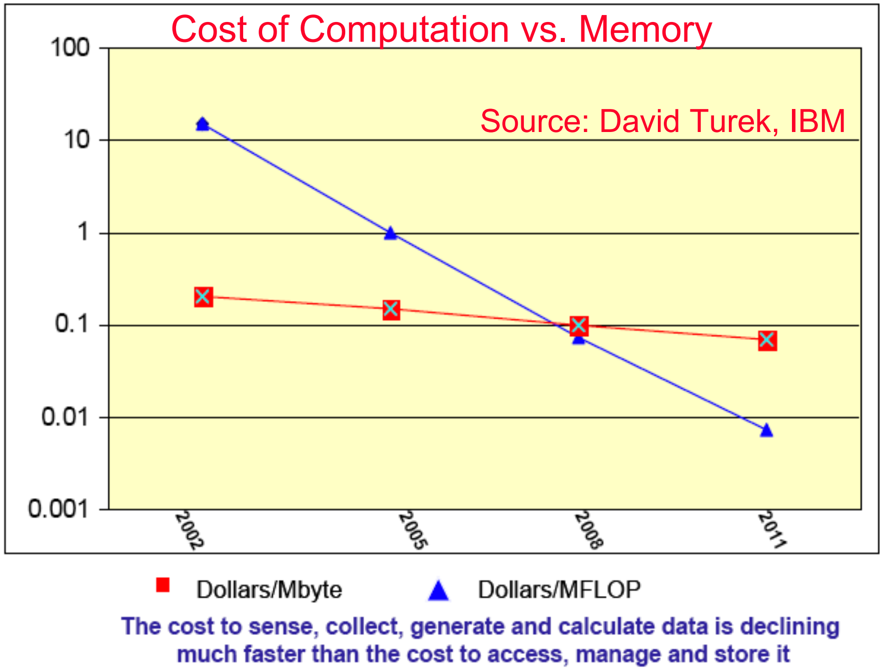
- Given enough parallel work, this is the biggest barrier to getting desired speedup
- Parallelism overheads include:
  - cost of starting a thread or process
  - cost of communicating shared data
  - cost of synchronizing
  - extra (redundant) computation
- Each of these can be in the range of milliseconds (=millions of flops) on some systems
- Tradeoff: Algorithm needs sufficiently large units of work to run fast in parallel (i.e. large granularity), but not so large that there is not enough parallel work

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

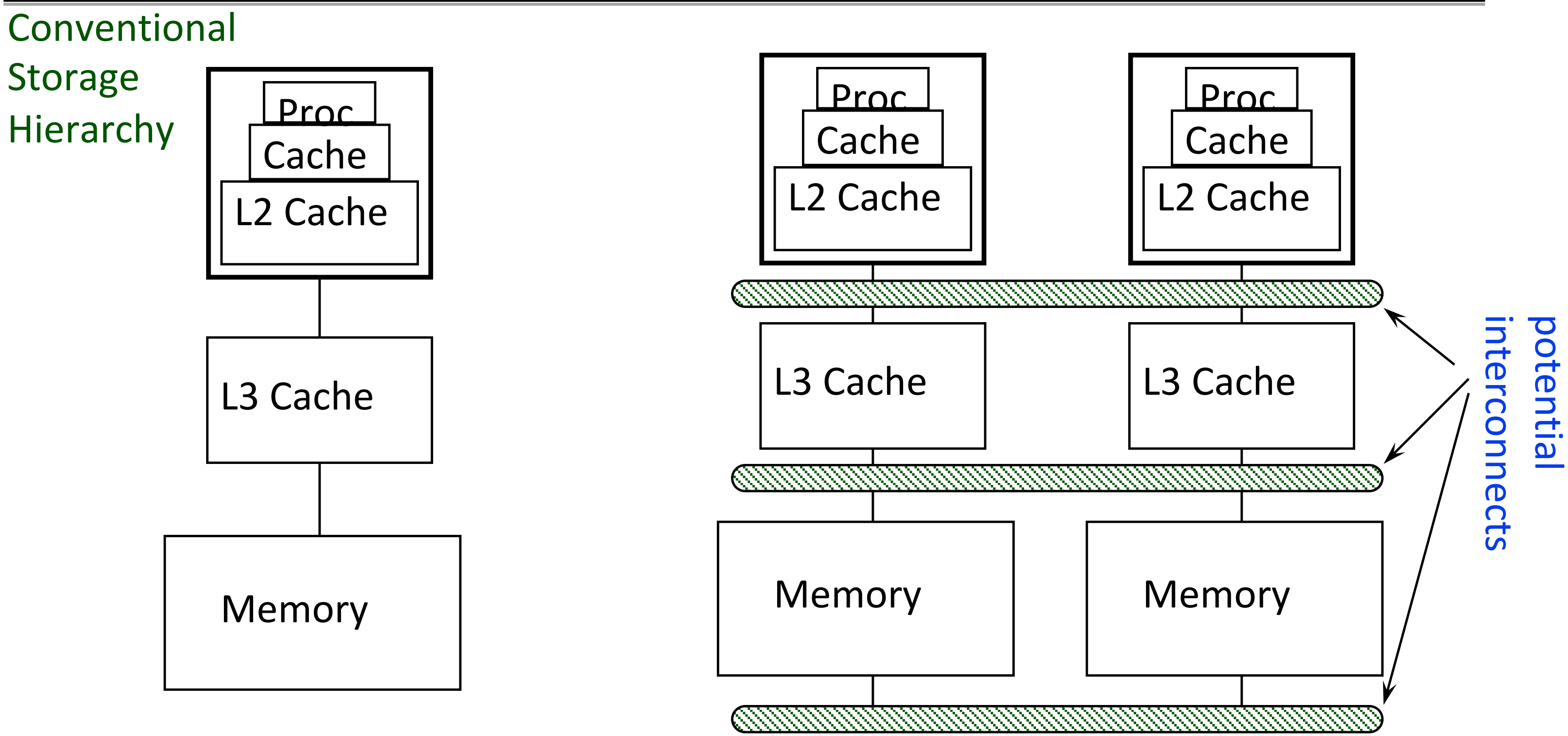




Question: Can you double concurrency without doubling memory?

- **Strong scaling**: fixed problem size, increase number of processors
- **Weak scaling**: grow problem size proportionally to number of processors

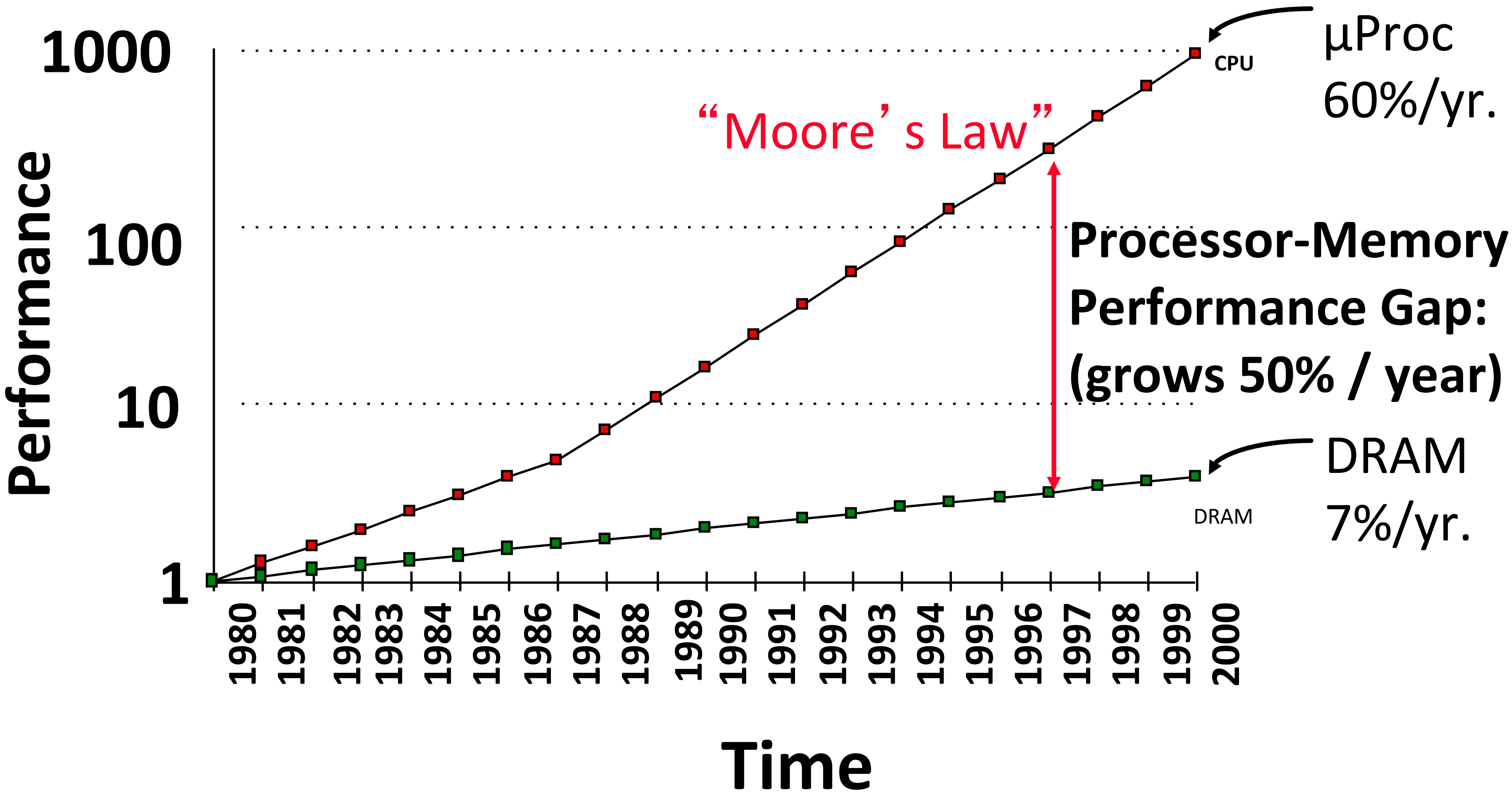
Locality and Parallelism



- Large memories are slow, fast memories are small
- Storage hierarchies are large and fast on average
- Parallel processors, collectively, have large, fast cache
  - the slow accesses to "remote" data we call "communication"
- Algorithm should do most work on local data

Processor-DRAM Gap (latency)

Goal: find algorithms that minimize communication, not necessarily arithmetic



Load Imbalance

- **Load imbalance is the time that some processors in the system are idle due to**
  - insufficient parallelism (during that phase)
  - unequal size tasks
- **Examples of the latter**
  - adapting to "interesting parts of a domain"
  - tree-structured computations
  - fundamentally unstructured problems
- **Algorithm needs to balance load**
  - Sometimes can determine work load, divide up evenly, before starting
    - "Static Load Balancing"
  - Sometimes work load changes dynamically, need to rebalance dynamically
    - "Dynamic Load Balancing," eg work-stealing

Parallel Software Eventually

- **2 types of programmers → 2 layers of software**
- **Efficiency Layer (20% of programmers)**
  - Expert programmers build Libraries implementing kernels, "Frameworks", OS, ….
  - Highest fraction of peak performance possible
- **Productivity Layer (80% of programmers)**
  - Domain experts / Non-expert programmers productively build parallel applications by composing frameworks & libraries
  - Hide as many details of machine, parallelism as possible
  - Willing to sacrifice some performance for productive programming
- **Expect students may want to work at either level**
  - In the meantime, we all need to understand enough of the efficiency layer to use parallelism effectively

Exercise

- Write a report on parallel computing

Thanks!