CSCI596: Scientific Computing & Visualization

Aiichiro Nakano

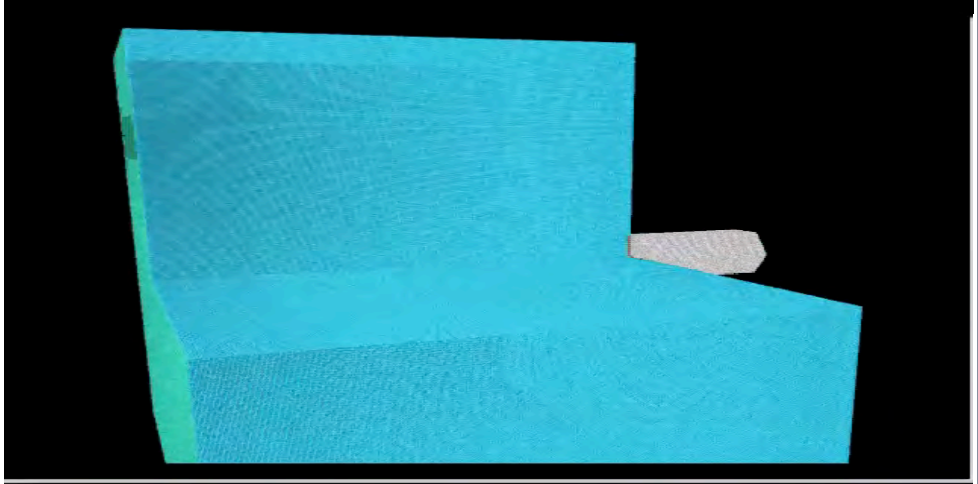
Collaboratory for Advanced Computing & Simulations
Department of Computer Science
Department of Physics & Astronomy
Department of Chemical Engineering & Materials Science
Department of Biological Sciences
University of Southern California

Email: anakano@usc.edu

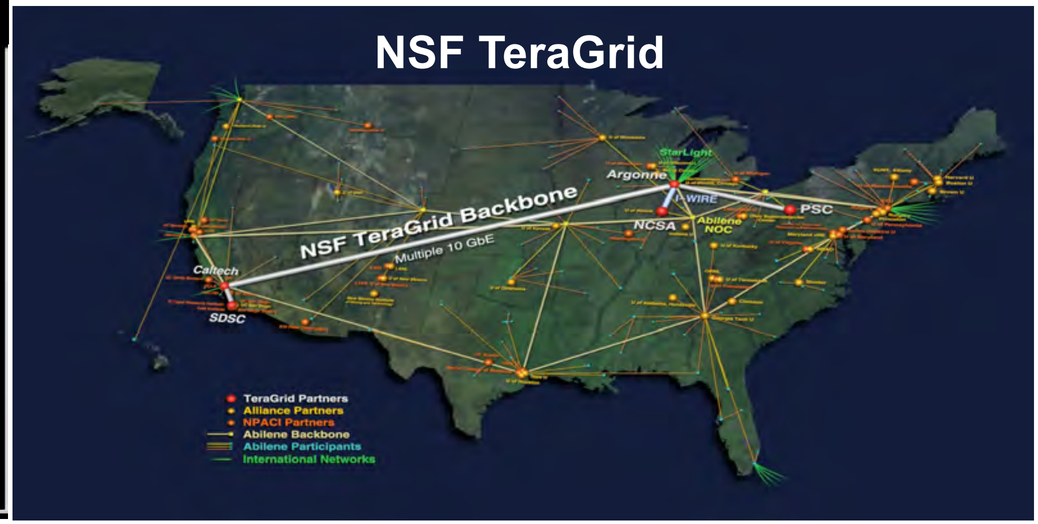
- **Computational science:** An area of scientific investigation, where computers play a central role—do science using computer (*e.g.*, computational chemistry).
- **Scientific computing:** An area in computer science (CS) to support computational sciences by innovative use of computer systems—parallel computing, scientific visualization, *etc*.

New Computing Architecture

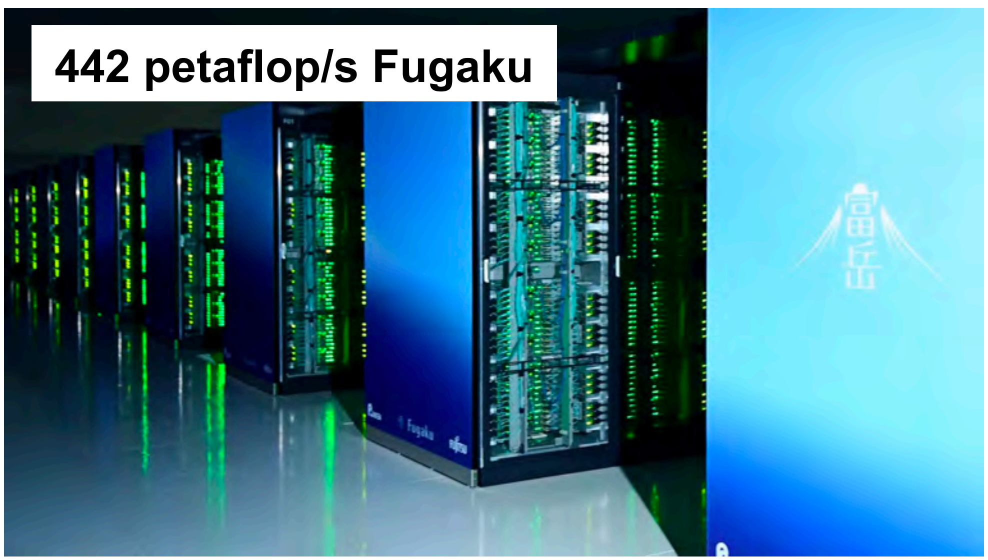
4.9 trillion-atom molecular dynamics
40 trillion-d.o.f. quantum mechanics



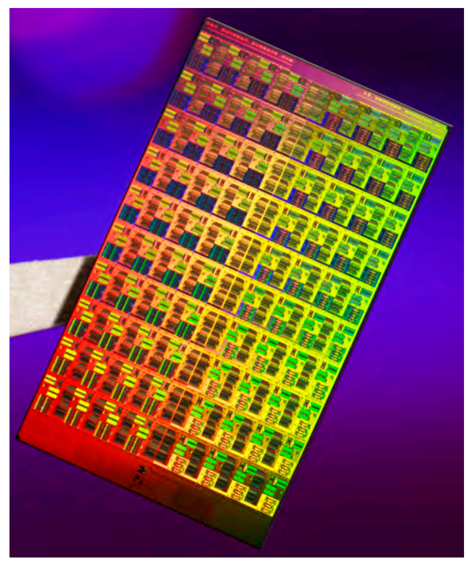
Global Grid of supercomputers



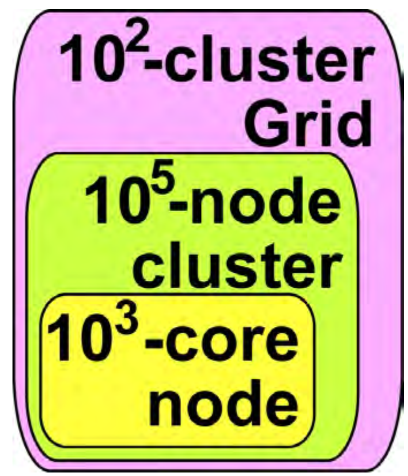
Toward exaflop/s computers



Many-core CPU computing

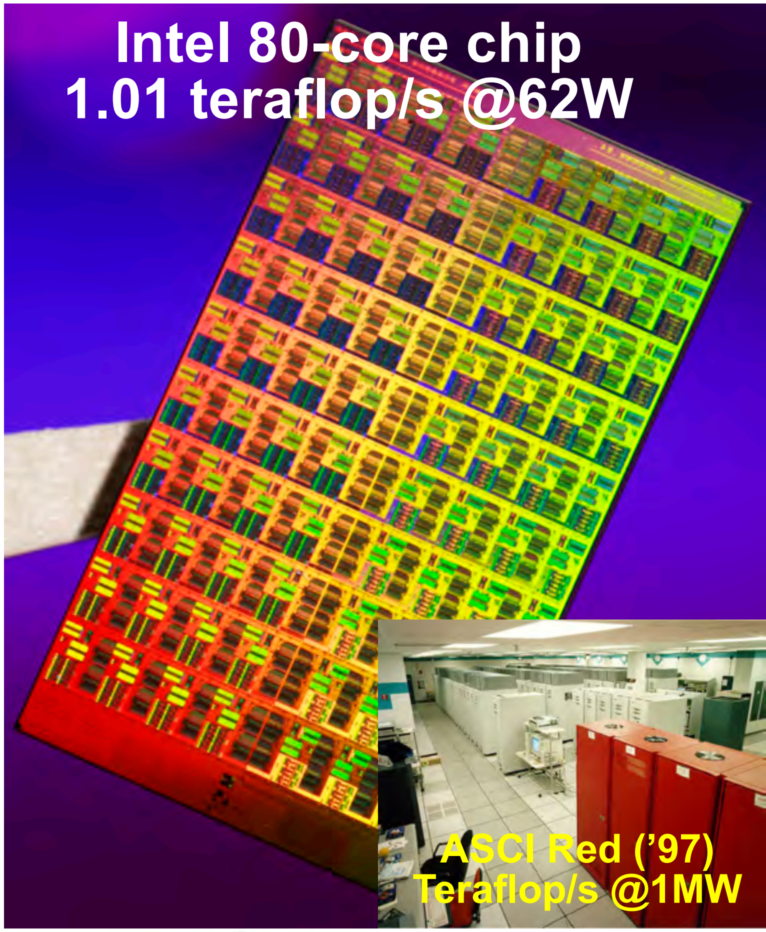


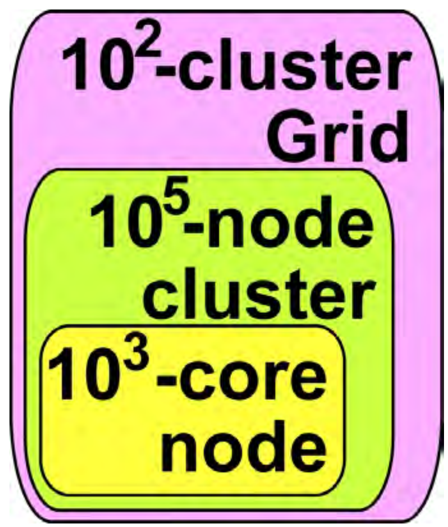
Intel 80-core chip
1Tflop/s@62W

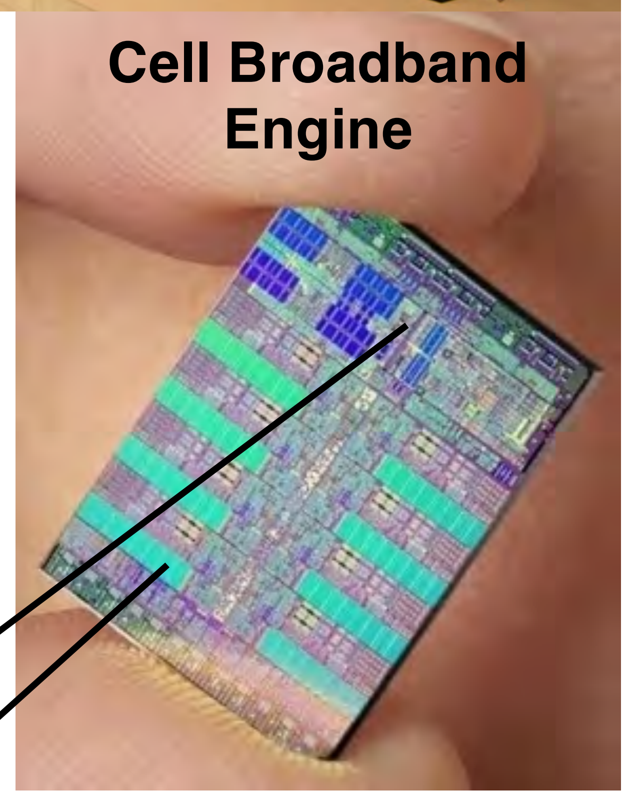


petaflop/s | exaflop/s = 10^{15} | 10^{18} floating-point operations per second

Many-core CPU/GPU Computing







6,912 cores
156 Tflop/s
NVIDIA Tesla A100

64bit PowerPC

8 synergistic processing elements

Godson-T Many-core Architecture

J. Parallel Distrib. Comput. 73 (2013) 1469–1482

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

ELSEVIER

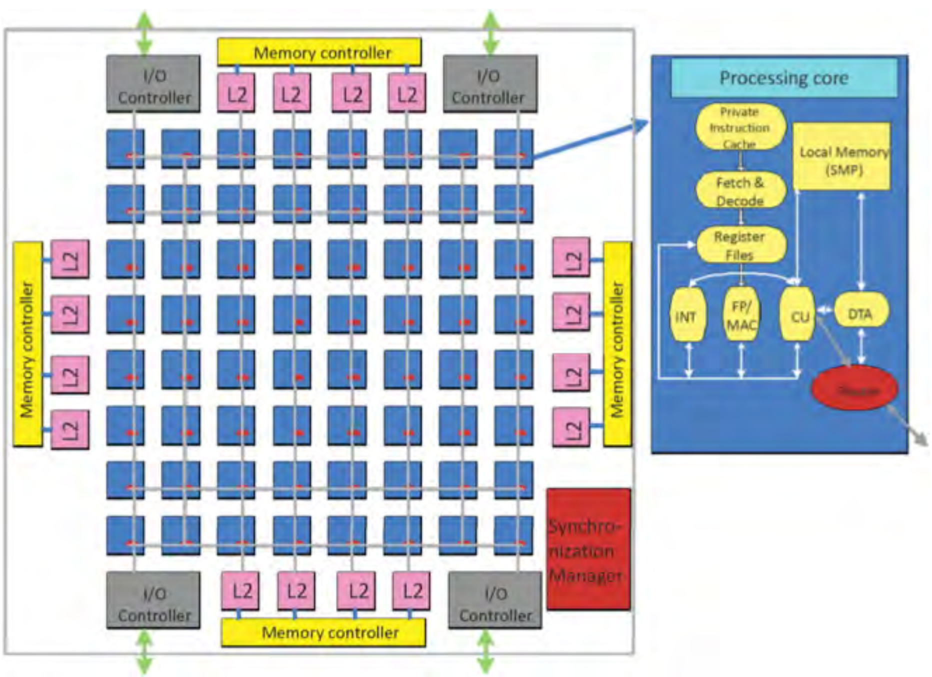
Scalability study of molecular dynamics simulation on *Godson-T* many-core architecture

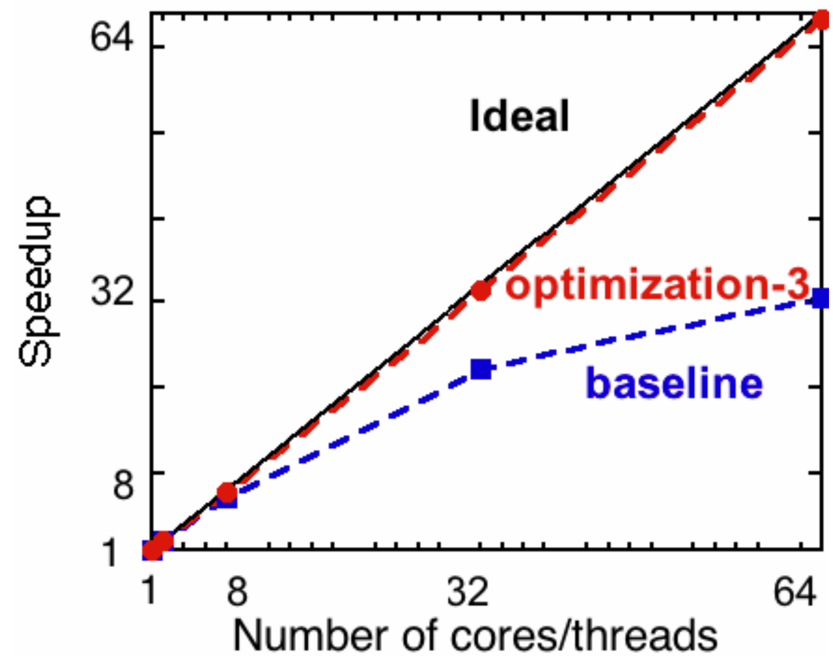
狗剩

Liu Peng [a,*], Guangming Tan [b,*], Rajiv K. Kalia [a], Aiichiro Nakano [a], Priya Vashishta [a], Dongrui Fan [b], Hao Zhang [b], Fenglong Song [b]

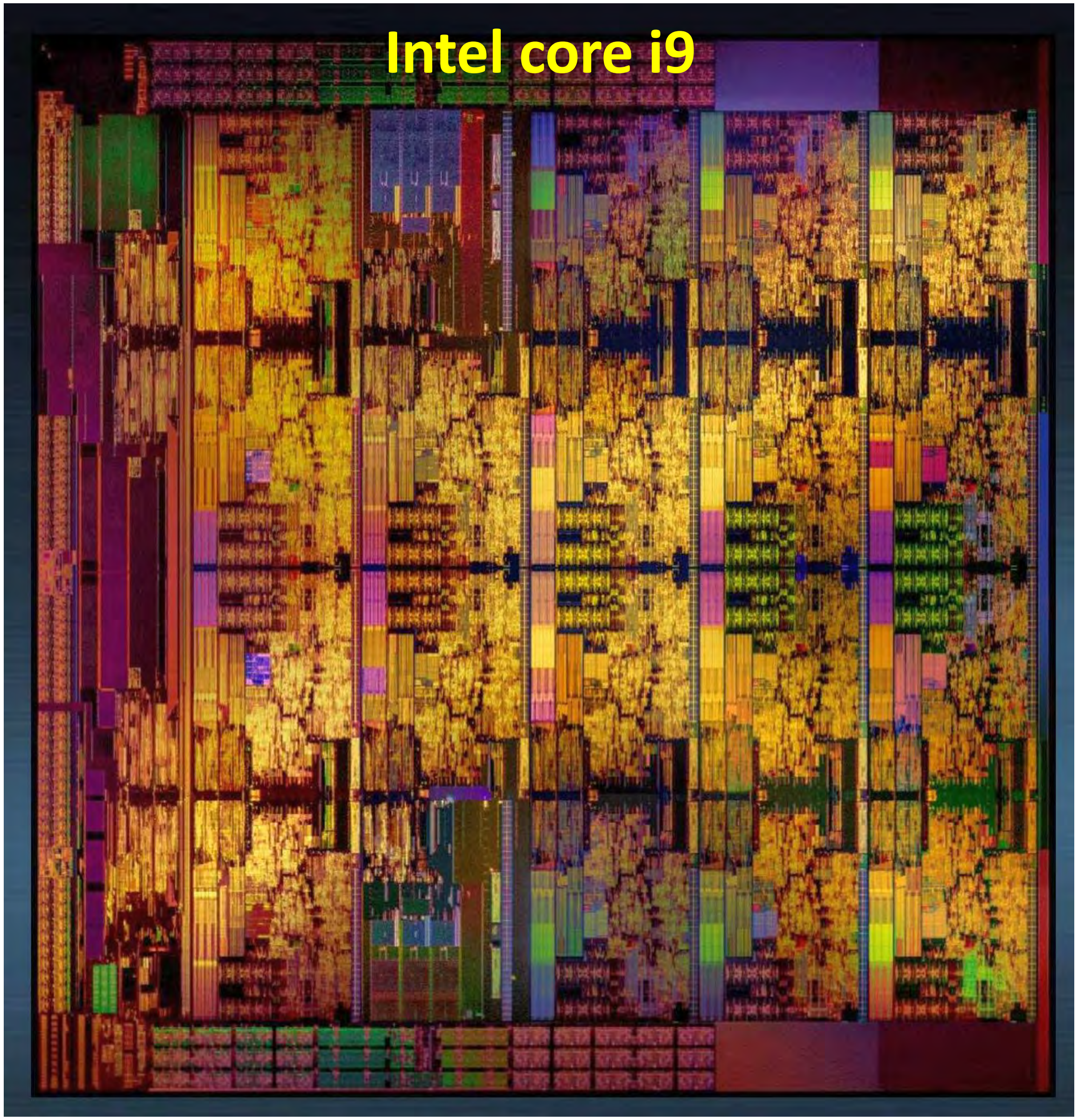
[a] Collaboratory for Advanced Computing and Simulations, University of Southern California, Los Angeles, CA, 90089, USA

[b] Key Laboratory of Computer System and Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China





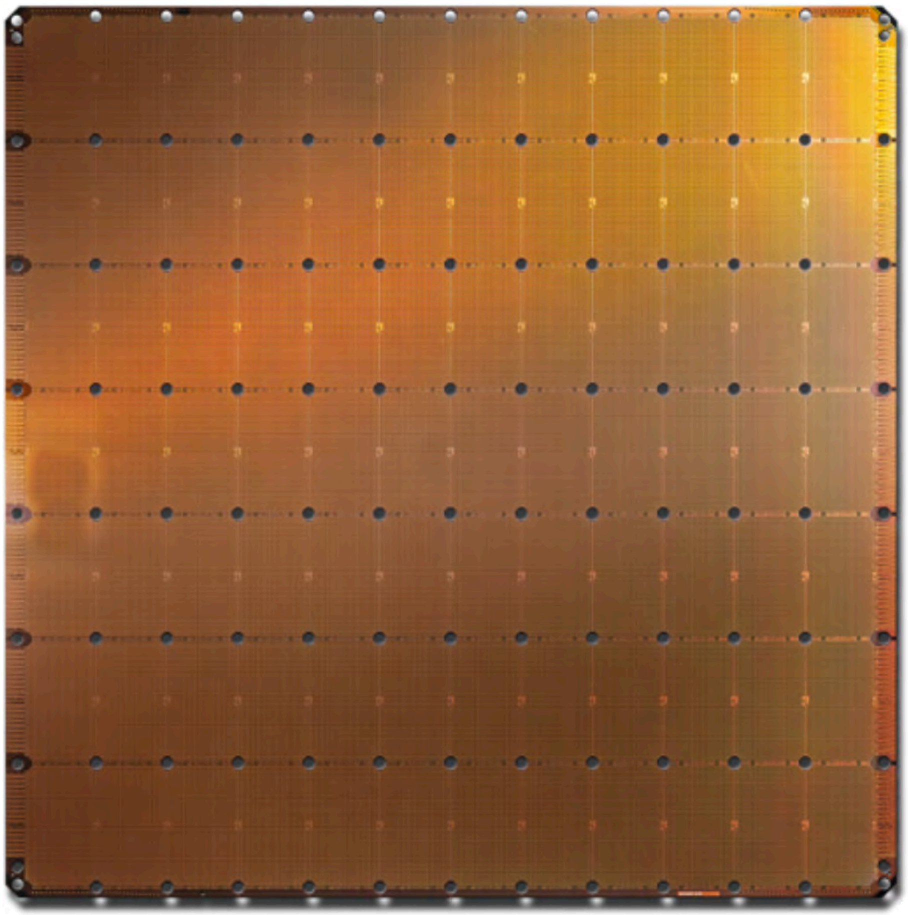
Multicore Processors

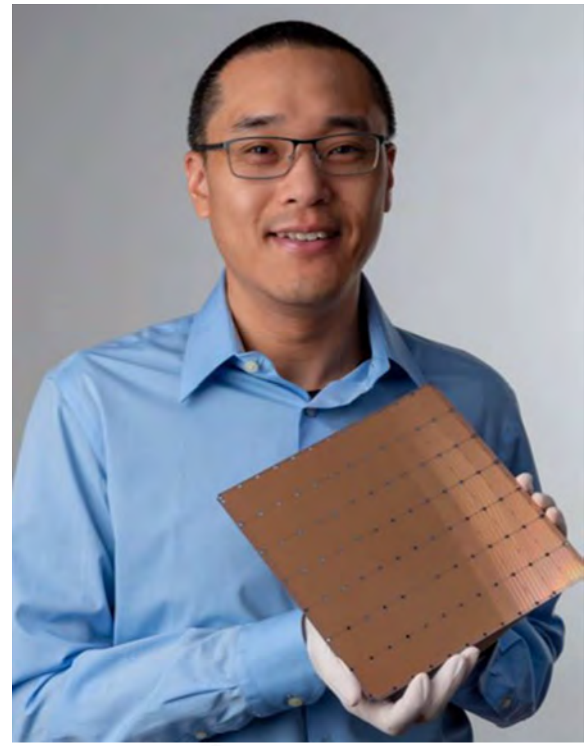


- **Multiple simple processors (or cores) sharing common memory**

World's Biggest Chip

Cerebras AI engine





Cerebras WSE

1.2 Trillion transistors
46,225 mm² silicon

Largest GPU

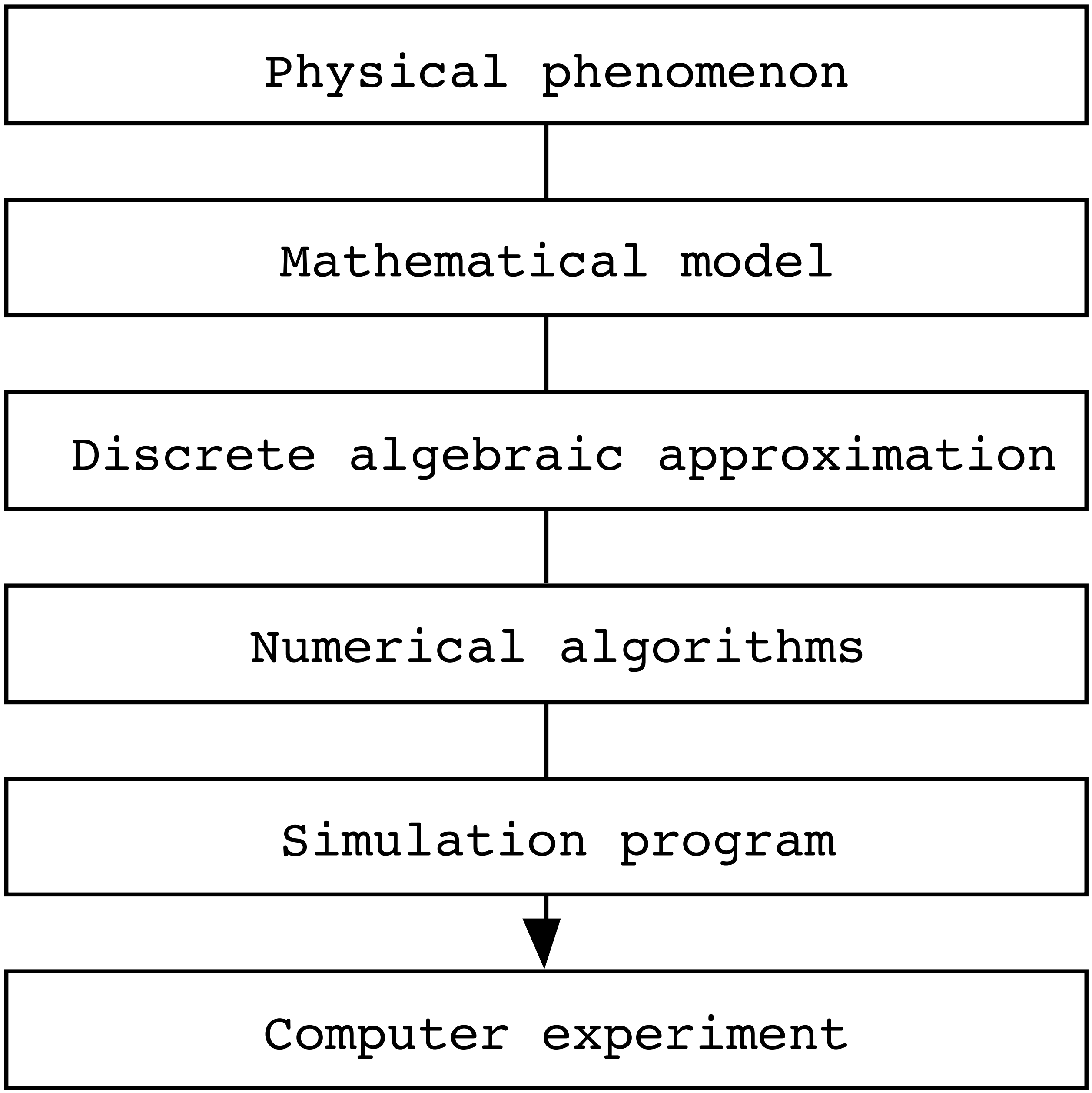
21.1 Billion transistors
815 mm² silicon

https://www.cerebras.net/

Computer Simulation

Run real applications on advanced computing architectures

Computer Experiment



Mathematical Model

PHILOSOPHIÆ
NATURALIS
PRINCIPIA
MATHEMATICA.

Autore J. S. NEWTON, *Trin. Coll. Cantab. Soc.* Mathefeos Profeffore *Lucafiano*, & Societatis Regalis Sodali.

IMPRIMATUR.
S. PEPYS, *Reg. Soc.* PRÆSES.
Julii 5. 1686.

LONDINI,

Juffu Societatis Regiæ ac Typis *Jofephi Streater*. Proftat apud plures Bibliopolas. *Anno* MDCLXXXVII.

TITLE PAGE OF THE FIRST EDITION OF THE PRINCIPIA
(See Appendix, Note 2, page 627)

AXIOMS, OR LAWS OF MOTION[1]

LAW I

Every body continues in its state of rest, or of uniform motion in a right line, unless it is compelled to change that state by forces impressed upon it.

PROJECTILES continue in their motions, so far as they are not retarded by the resistance of the air, or impelled downwards by the force of gravity. A top, whose parts by their cohesion are continually drawn aside from rectilinear motions, does not cease its rotation, otherwise than as it is retarded by the air. The greater bodies of the planets and comets, meeting with less resistance in freer spaces, preserve their motions both progressive and circular for a much longer time.

LAW II[2]

The change of motion is proportional to the motive force impressed; and is made in the direction of the right line in which that force is impressed.

If any force generates a motion, a double force will generate double the motion, a triple force triple the motion, whether that force be impressed altogether and at once, or gradually and successively. And this motion (being always directed the same way with the generating force), if the body moved before, is added to or subtracted from the former motion, according as they directly conspire with or are directly contrary to each other; or obliquely joined, when they are oblique, so as to produce a new motion compounded from the determination of both.

LAW III

To every action there is always opposed an equal reaction: or, the mutual actions of two bodies upon each other are always equal, and directed to contrary parts.

Whatever draws or presses another is as much drawn or pressed by that other. If you press a stone with your finger, the finger is also pressed by the

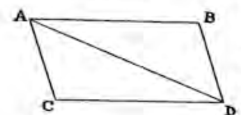
[1 Appendix, Note 14.] [2 Appendix, Note 15.]

stone. If a horse draws a stone tied to a rope, the horse (if I may so say) will be equally drawn back towards the stone; for the distended rope, by the same endeavor to relax or unbend itself, will draw the horse as much towards the stone as it does the stone towards the horse, and will obstruct the progress of the one as much as it advances that of the other. If a body impinge upon another, and by its force change the motion of the other, that body also (because of the equality of the mutual pressure) will undergo an equal change, in its own motion, towards the contrary part. The changes made by these actions are equal, not in the velocities but in the motions of bodies; that is to say, if the bodies are not hindered by any other impediments. For, because the motions are equally changed, the changes of the velocities made towards contrary parts are inversely proportional to the bodies. This law takes place also in attractions, as will be proved in the next Scholium.

COROLLARY I

A body, acted on by two forces simultaneously, will describe the diagonal of a parallelogram in the same time as it would describe the sides by those forces separately.

If a body in a given time, by the force M impressed apart in the place A, should with an uniform motion be carried from A to B, and by the force N impressed apart in the same place, should be carried from A to C, let the parallelogram ABCD be completed, and, by both forces acting together, it will in the same time be carried in the diagonal from A to D. For since the force N acts in the direction of the line AC, parallel to BD, this force (by the second Law) will not at all alter the velocity generated by the other force M, by which the body is carried towards the line BD. The body therefore will arrive at the line BD in the same time, whether the force N be impressed or not; and therefore at the end of that time it will be found somewhere in the line BD. By the same argument, at the end of the same time it will be found somewhere in the line CD. Therefore it will be found in the point D, where both lines meet. But it will move in a right line from A to D, by Law I.



LAW I

Every body continues in its state of rest, or of uniform motion in a right line, unless it is compelled to change that state by forces impressed upon it.

LAW II[2]

The change of motion is proportional to the motive force impressed; and is made in the direction of the right line in which that force is impressed.

Towards 2020 Science

Some 400 years later, Newton, in his efforts to understand the natural laws of the rate of change in motion, used algebra to underpin another new branch of mathematics: calculus (a branch for which von Leibniz is simultaneously and independently credited). Calculus spurred scientists "to go off looking for other laws of nature that could explain natural phenomenon in terms of rates of change and found them by the bucketful - heat, sound, light, fluid dynamics, electricity and magnetism" [2].

A scientific revolution is just beginning.

However, what this report uncovers, for the first time, is a fundamentally important shift from *computers* supporting scientists to 'do' traditional science to *computer science* becoming embedded into the very fabric of science and how science is done, creating what I am prepared to go so far as to call 'new kinds' of science[1].

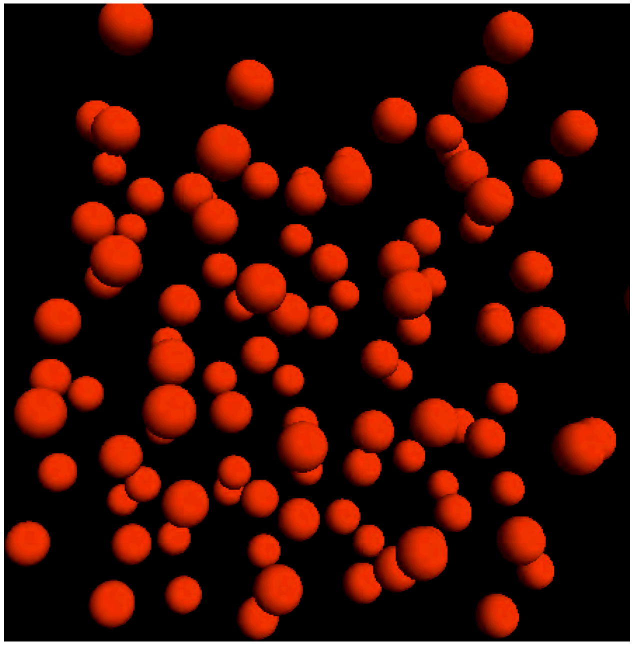
Stephen Emmott

http://research.microsoft.com/towards2020science
http://www.nature.com/nature/journal/v440/n7083

Molecular Dynamics Simulation

- **Newton's equation of motion**

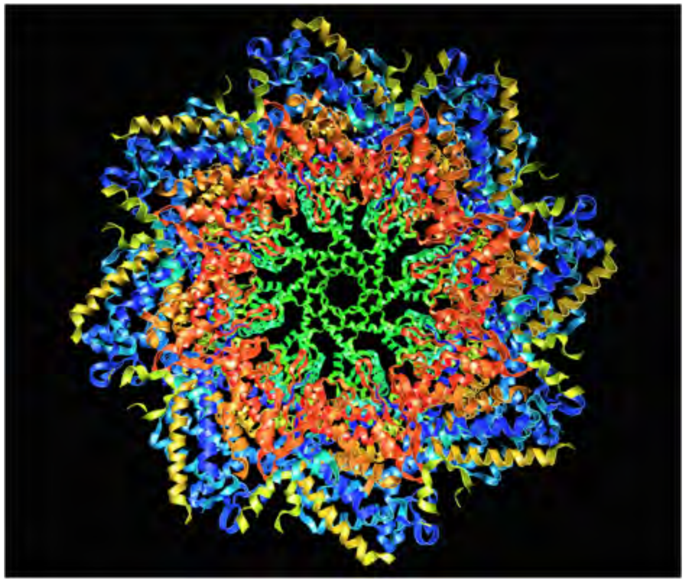
$$m_i \frac{d^2\mathbf{r}_i}{dt^2} = -\frac{\partial V(\mathbf{r}^N)}{\partial \mathbf{r}_i} \quad (i = 1,\ldots,N)$$

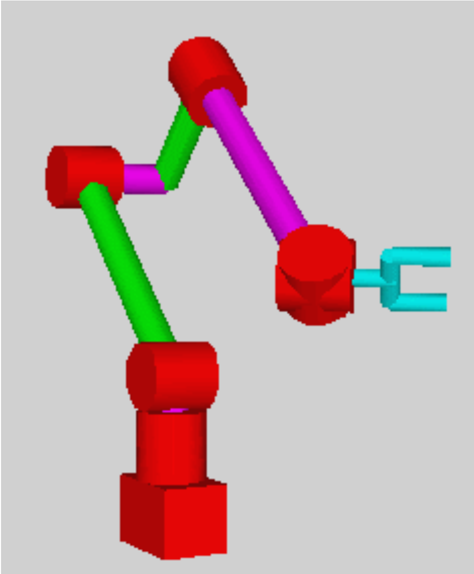


- **Many-body interatomic potential**

$$V = \sum_{i<j} u_{ij}(|\mathbf{r}_{ij}|) + \sum_{i,j<k} v_{jik}(\mathbf{r}_{ij}, \mathbf{r}_{ik})$$

- **Application: drug design, robotics, entertainment, *etc*.**

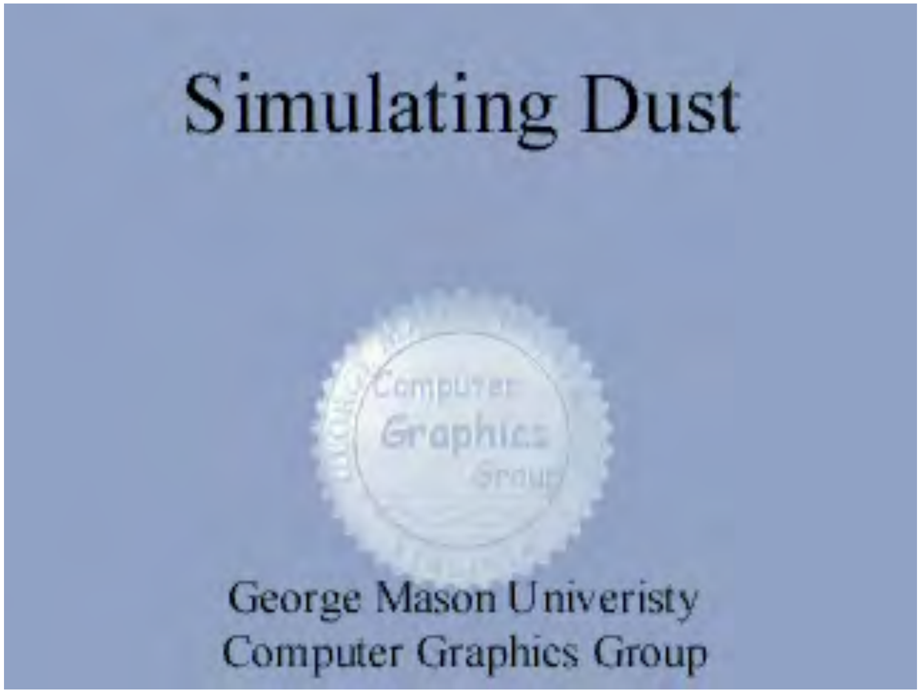




A scene from the movie "Twister"

Molecular Dynamics in Graphics

Particle modeling of dust & smoke



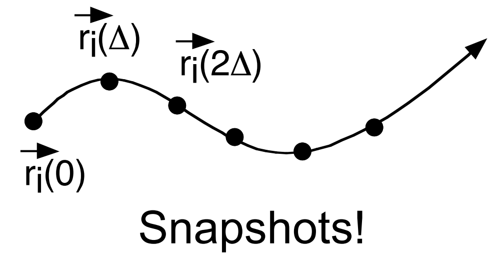
Jim Chen (George Mason)

Ron Fedkiw (Stanford)

MD Algorithm

Time discretization: differential → algebraic equation

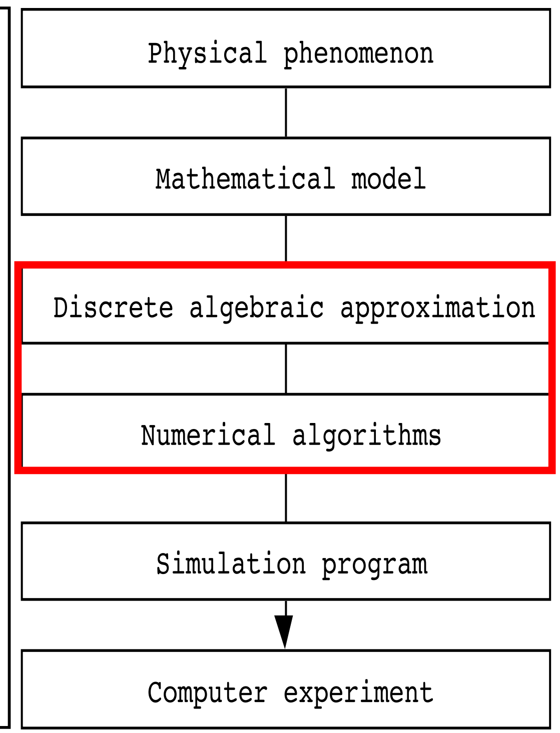
$$\begin{cases} \vec{r}_i(t+\Delta) = \vec{r}_i(t) + \vec{v}_i(t)\Delta + \dfrac{1}{2}\vec{a}_i(t)\Delta^2 \\ \vec{v}_i(t+\Delta) = \vec{v}_i(t) + \dfrac{\vec{a}_i(t) + \vec{a}_i(t+\Delta)}{2}\Delta \end{cases} \qquad \vec{a}_i = -\frac{1}{m}\frac{\partial V}{\partial \vec{r}_i}$$



Time stepping: Velocity Verlet algorithm

Given $\left(\vec{r}_i(t), \vec{v}_i(t)\right)$

1. Compute $\vec{a}_i(t)$ as a function of $\{\vec{r}_i(t)\}$
2. $\vec{v}_i\left(t+\dfrac{\Delta}{2}\right) \leftarrow \vec{v}_i(t) + \dfrac{\Delta}{2}\vec{a}_i(t)$
3. $\vec{r}_i(t+\Delta) \leftarrow r_i(t) + \vec{v}_i\left(t+\dfrac{\Delta}{2}\right)\Delta$
4. Compute $\vec{a}_i(t+\Delta)$ as a function of $\{\vec{r}_i(t+\Delta)\}$
5. $\vec{v}_i(t+\Delta) \leftarrow \vec{v}_i\left(t+\dfrac{\Delta}{2}\right) + \dfrac{\Delta}{2}\vec{a}_i(t+\Delta)$



MD Program

```
for (n=0; n<nAtom; n++)
  for (k=0; k<3; k++)
    rv[n][k] = rv[n][k] + DeltaT/2*ra[n][k];
```

$$\vec{v}_i(t+\frac{\Delta}{2}) \leftarrow \vec{v}_i(t) + \frac{\Delta}{2}\vec{a}_i(t)$$

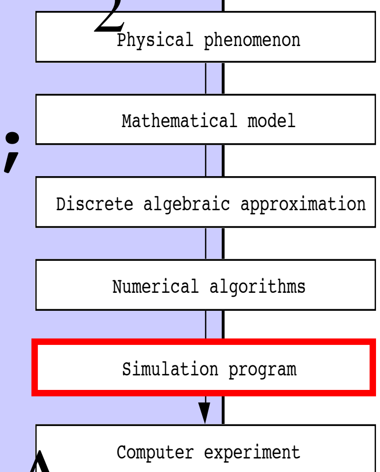
```
for (n=0; n<nAtom; n++)
  for (k=0; k<3; k++)
    r[n][k] = r[n][k] + DeltaT*rv[n][k];
```

$$\vec{r}_i(t+\Delta) \leftarrow \vec{r}_i(t) + \vec{v}_i(t+\frac{\Delta}{2})\Delta$$

```
ComputeAccel();    // r[][] → ra[][]
```

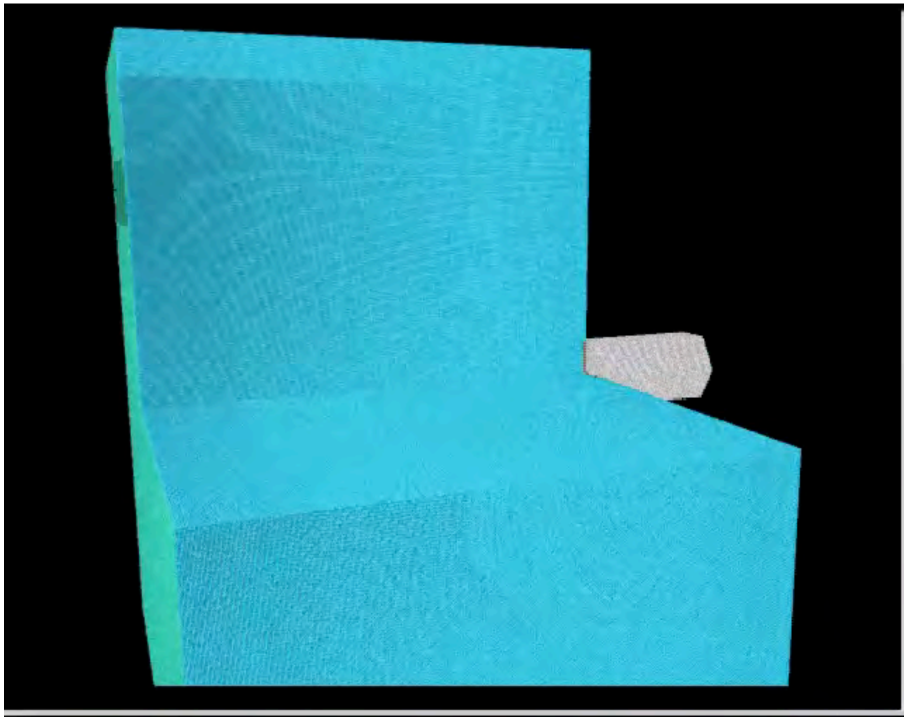
```
for (n=0; n<nAtom; n++)
  for (k=0; k<3; k++)
    rv[n][k] = rv[n][k] + DeltaT/2*ra[n][k];
```

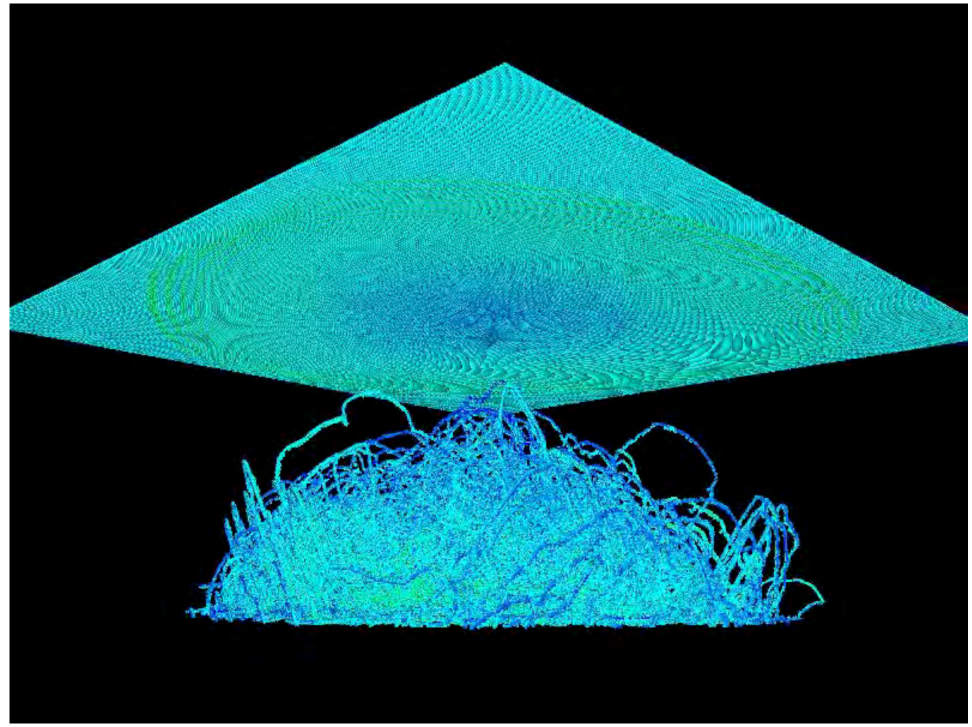
$$\vec{v}_i(t+\frac{\Delta}{2}) \leftarrow \vec{v}_i(t) + \frac{\Delta}{2}\vec{a}_i(t)$$

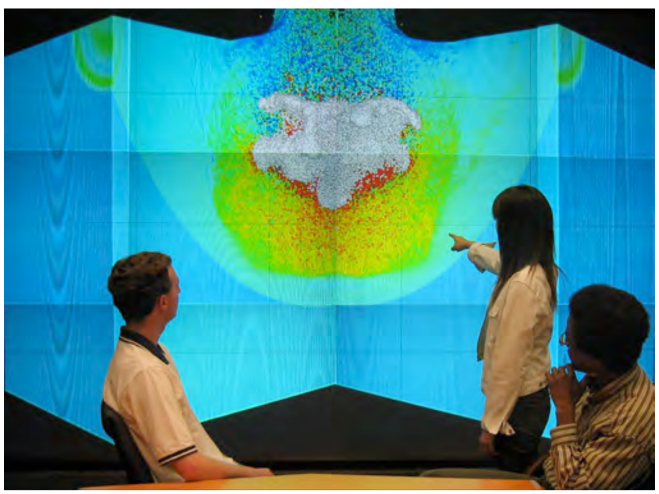


Computer Experiment

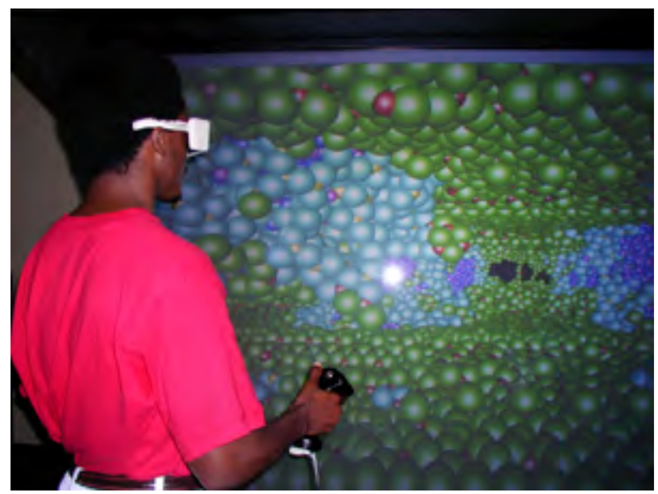
200-500 million atom molecular dynamics simulation of hypervelocity (15 km/s) impact on AlN & Al_2O_3 plate







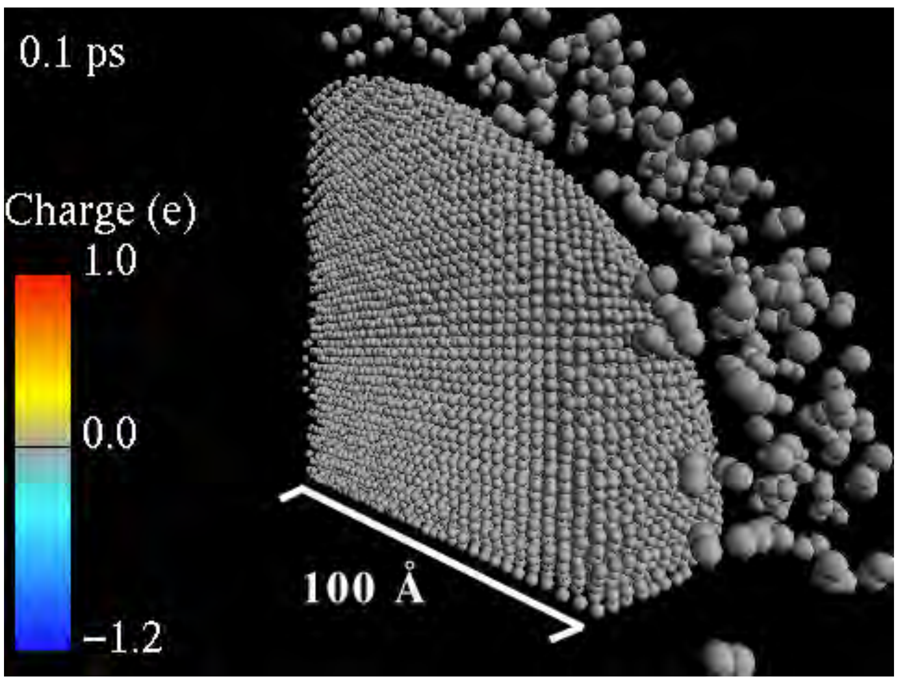
Physical phenomenon
↓
Mathematical model
↓
Discrete algebraic approximation
↓
Numerical algorithms
↓
Simulation program
↓
Computer experiment



Type of Mathematical Models

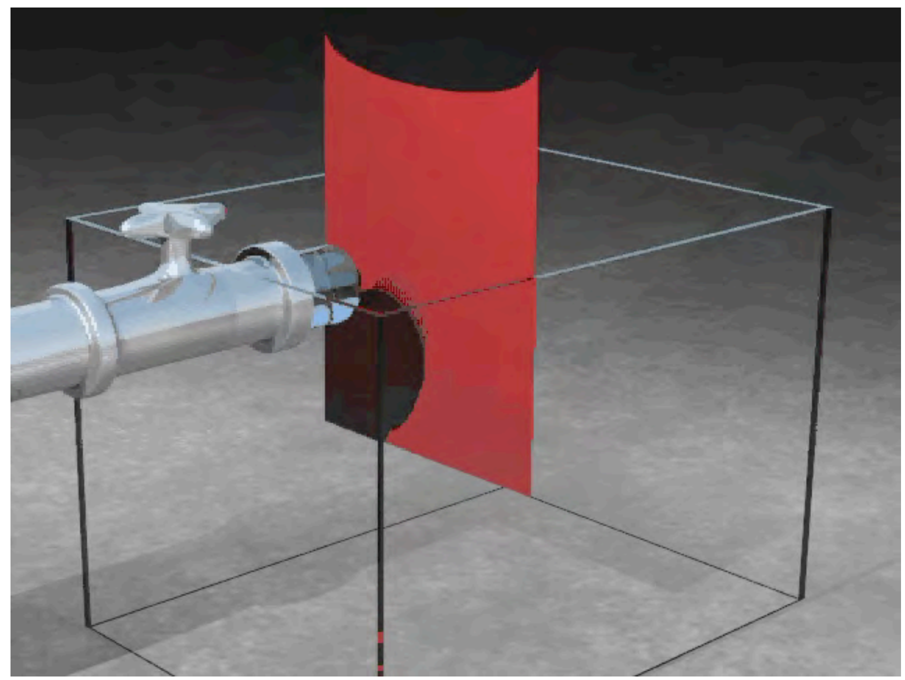
	Particle model (ordinary differential equations)	Continuum model (partial differential equations)
Deterministic	molecular dynamics	computational fluid dynamics, continuum mechanics
Stochastic	Monte Carlo particle simulation	quantum Monte Carlo

Particle model of oxidation



Tim Campbell
USC-CACS

Continuum model of water flow



Ron Fedkiw (Stanford)
graphics.stanford.edu/~fedkiw

Continuum Model: Quantum Mechanics

Challenge: Complexity of quantum *N*-body problem

Density functional theory (DFT)

(Walter Kohn, Nobel Chemistry Prize, '98)

$$\psi(\mathbf{r}_1, \ldots, \mathbf{r}_{N\mathrm{el}}) \qquad O(C^N)$$

$$\longrightarrow$$

$$\{\psi_n(\mathbf{r}) | n = 1, \ldots, N_{\mathrm{el}}\} \qquad O(N^3)$$

Constrained minimization problem:

Minimize:

$$E[\{\psi_n\}] = \sum_{n=1}^{N_{\mathrm{el}}} \int d\mathbf{r}\, \psi_n^*(\mathbf{r}) \left(-\frac{\hbar^2}{2m_{\mathrm{el}}} \frac{\partial^2}{\partial \mathbf{r}^2} + V_{\mathrm{ion}}(\mathbf{r}) \right) \psi_n(\mathbf{r}) + \frac{e^2}{2} \iint d\mathbf{r} d\mathbf{r}' \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} + E_{\mathrm{xc}}[\rho(\mathbf{r})]$$

with orthonormal constraints: $\int d^3 r \psi_m^*(\mathbf{r}) \psi_{\mathrm{n}}(\mathbf{r}) = \delta_{mn}$

$$\text{Charge density: } \rho(\mathbf{r}) = \sum_{n=1}^{N_{\mathrm{el}}} |\psi_{\mathrm{n}}(\mathbf{r})|^2$$

DFT: It's Algorithm

Density functional theory (DFT)

Hohenberg & Kohn, *Phys. Rev*. **136**, B864 ('64); W. Kohn [*Nobel chemistry prize*, '98]

$O(C^N)$ → $O(N^3)$

1 N-electron problem → **N 1-electron problems**

AIP | The Journal of Chemical Physics

Volume 140, Issue 18, 14 May 2014

SPECIAL TOPIC: ADVANCES IN DENSITY FUNCTIONAL THEORY

A divide-conquer-recombine algorithmic paradigm for large spatiotemporal quantum molecular dynamics simulations

Fuyuki Shimojo[1,2], Shinnosuke Hattori[1,2], Rajiv K. Kalia[1], Manaschai Kunaseth[1,3], Weiwei Mou[1], Aiichiro Nakano[1], Ken-ichi Nomura[1], Satoshi Ohmura[1,2,4], Pankaj Rajak[1], Kohei Shimamura[1,2,5] and Priya Vashishta[1]

+ VIEW AFFILIATIONS

J. Chem. Phys. **140**, 18A529 (2014); http://dx.doi.org/10.1063/1.4869342

$O(N^3)$ → $O(N)$

Mean-field theory → **Divide-conquer-recombine**

Divide-Conquer-(Re)combine

- "The first was to never accept anything as true which I could not accept as obviously true. The second was to divide each of the problems in as many parts as I should to solve them. The third, beginning with the simplest and easiest to understand matters, little by little, to the most complex knowledge. And the last resolution was to make my enumerations so complete and my reviews so general that I could be assured that I had not omitted anything." (René Descartes, *Discourse on Method*, 1637)

divide (conquer) recombine

- 「モデルの分割ー再統合の方法の優れた点は、分割した要素的概念を、モデルの理解に役立つように再構成することができ、そこに創造の入り込む余地があるという点にある。」（福井謙一 学問の創造、１９８７）

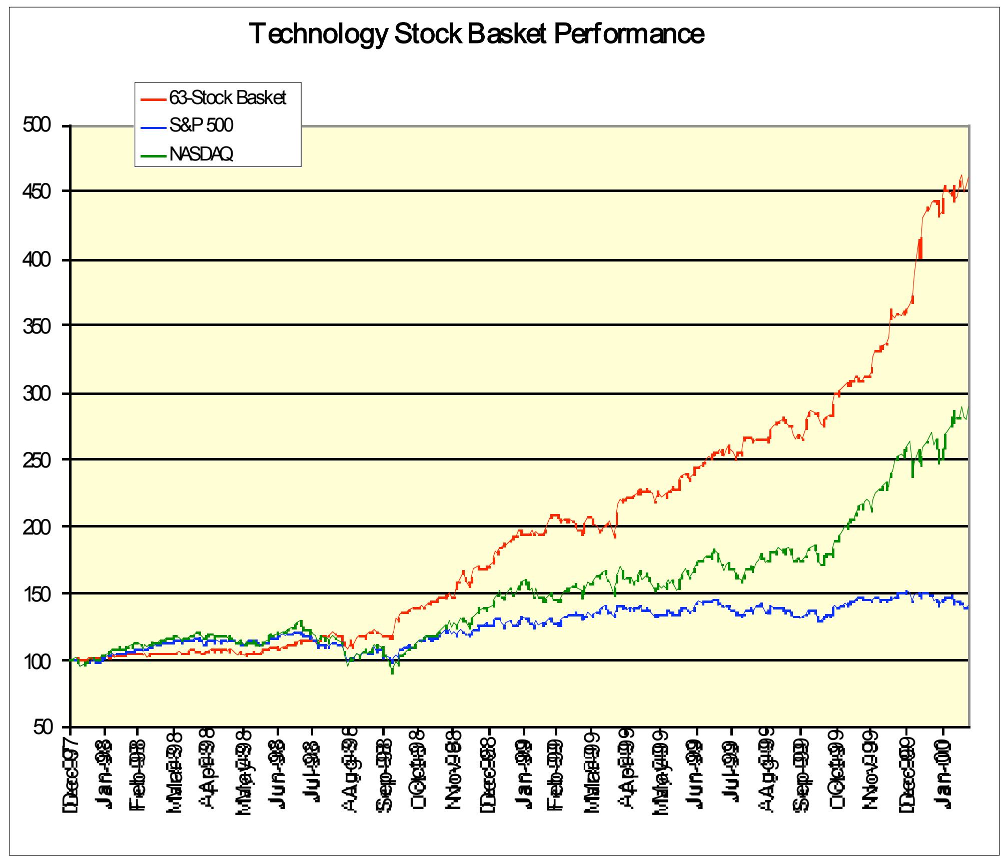
room for creativity

Kenichi Fukui [Nobel Chemistry Prize, '98]

Stochastic Model of Stock Prices

Fluctuation in stock price

Computational stock portfolio trading



Andrey Omeltchenko (Quantlab)

Basis of Black-Scholes analysis of option prices

$$dS = \mu S dt + \sigma S \varepsilon \sqrt{dt}$$

(1997 Nobel Economy Prize to Myron Scholes)

First Gauss Prize

The International Mathematical Union (IMU) and
the Deutsche Mathematiker-Vereinigung (DMV)
jointly award the
Carl Friedrich Gauss Prize for Applications of Mathematics
to **Professor Dr. Kiyoshi Itô**

for laying the foundations of the Theory of Stochastic Differential Equations and Stochastic Analysis. Itô's work has emerged as one of the major mathematical innovations of the 20th century and has found a wide range of applications outside of mathematics. Itô calculus has become a key tool in areas such as engineering (e.g., filtering, stability, and control in the presence of noise), physics (e.g., turbulence and conformal field theory), and biology (e.g., population dynamics). It is at present of particular importance in economics and finance with option pricing as a prime example.

Madrid, August 22, 2006

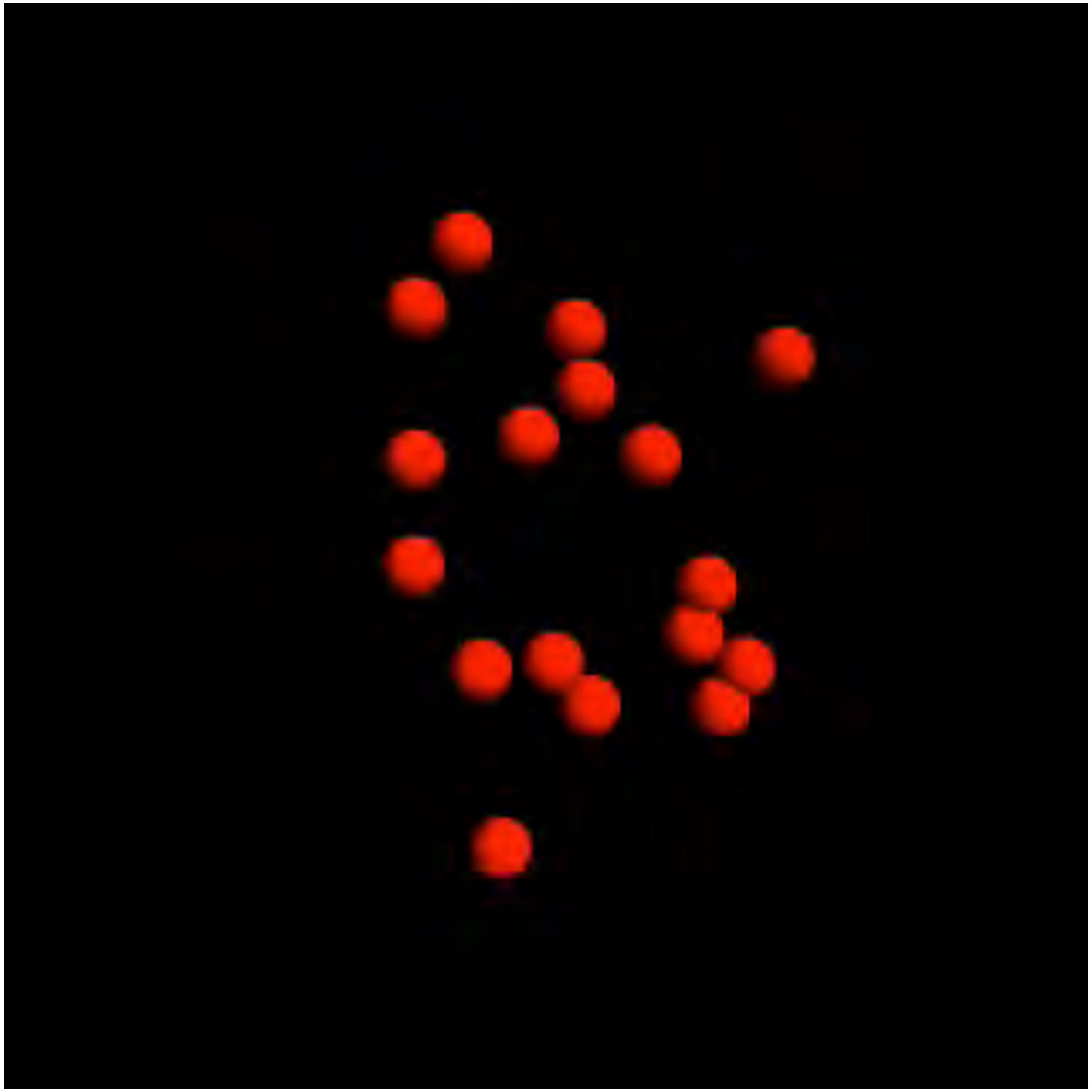
Sir John Ball
President of IMU

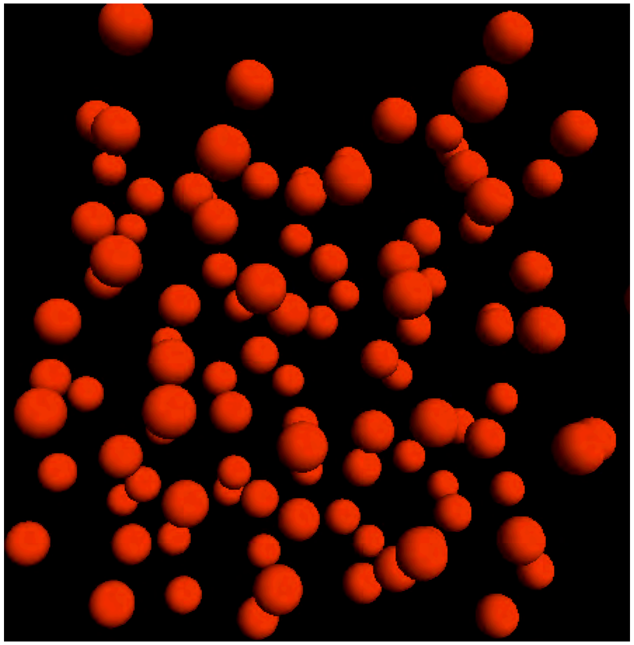
$$dS = \mu S dt + \sigma S \varepsilon \sqrt{dt}$$

Günter M. Ziegler
President of DMV

Martin Grötschel

Monte Carlo Simulation

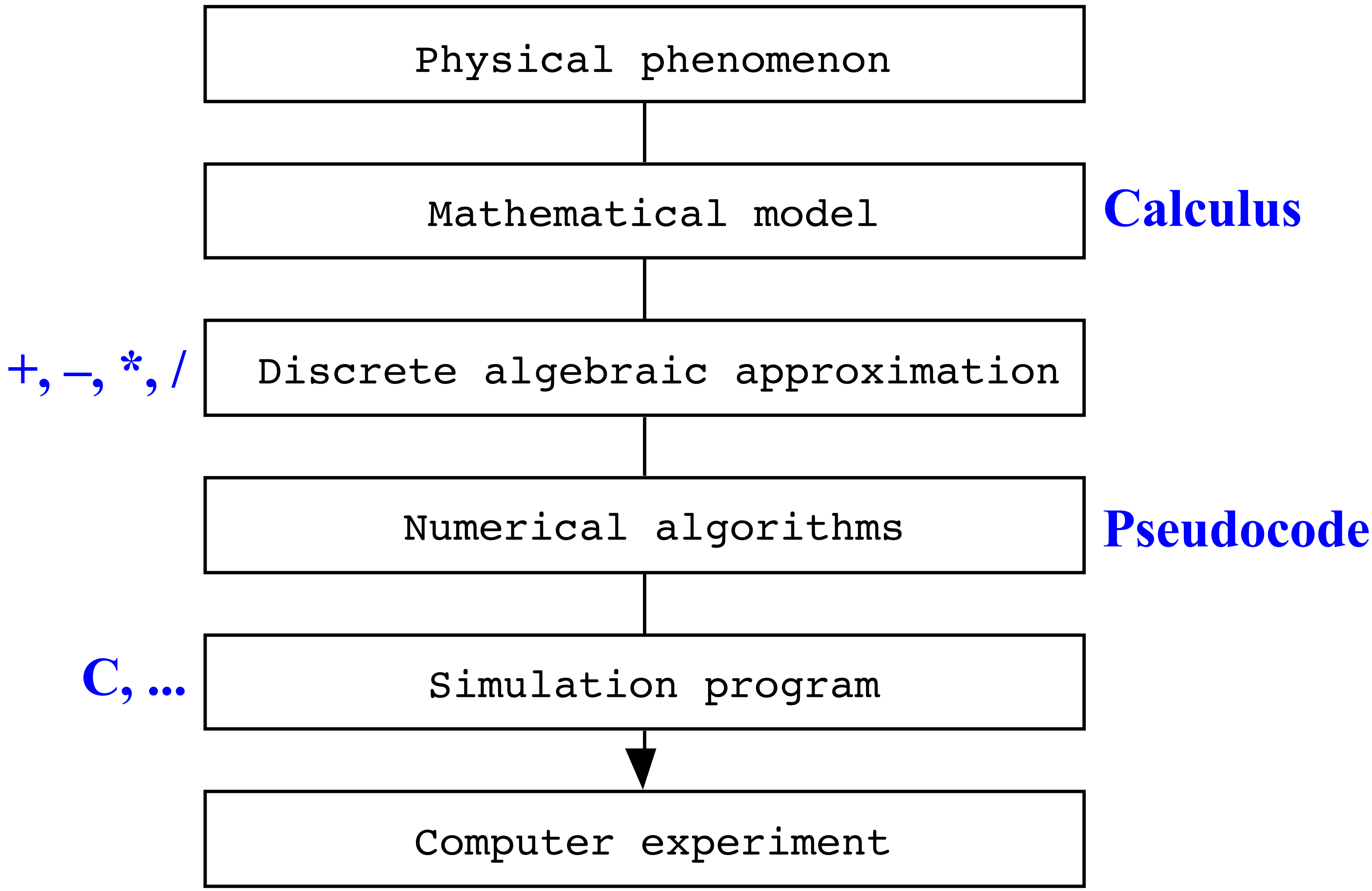




cf. Molecular dynamics

- **Random trial → acceptance by a cost criterion**
- **Combinatorial optimization by simulated annealing**

Recap: Scientific Computing



- **Particle simulation as an archetypal example**
- **Use your own application in the final project**

History of Particle Simulations

’44 John von Neumann memo on a stored-program computer: *“Our present analytical methods seem unsuitable for the solution of the important problems arising in connection with nonlinear partial differential equations. The really efficient high-speed computing devices may provide us with those heuristic hints which are needed in all parts of mathematics for genuine progress”*

’53 First Monte Carlo simulation of liquid by Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller on MANIAC at Los Alamos Nat’l Lab

’55 Enrico Fermi, John Pasta, and Stanislaw Ulam studied the dynamics of an one-dimensional array of particles coupled by anharmonic springs on MANIAC

’56 Dynamics of hard spheres (billiards) studied by Alder and Wainwright at the Lawrence Livermore Nat’l Lab.

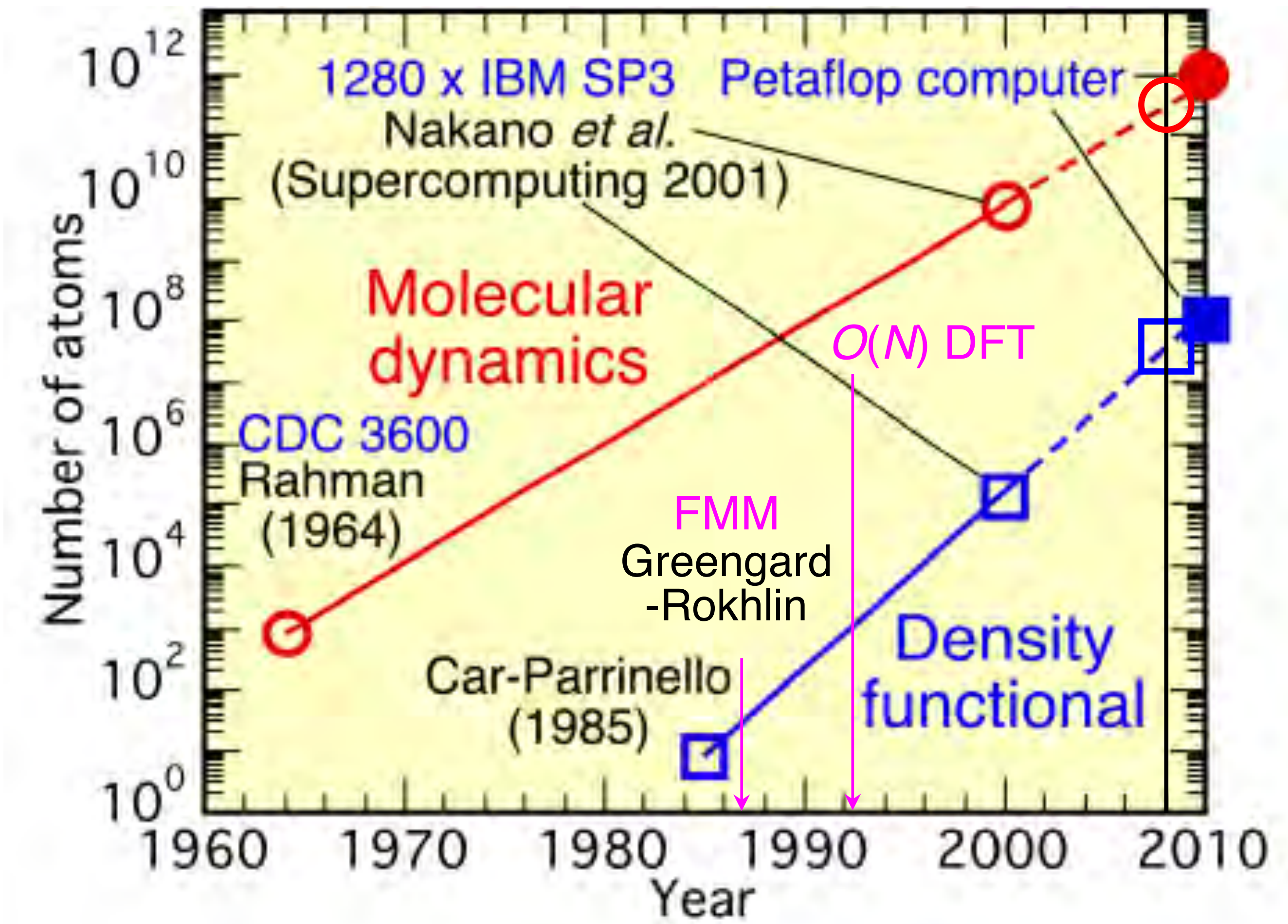
’60 Radiation damage in crystalline Cu studied with short-range repulsion and uniform attraction toward the center by George Vineyard’s group at Brookhaven Nat’l Lab

’64 First MD simulation of liquid (864 argon atoms) using interatomic potentials by Aneesur Rahman at the Argonne Nat’l Lab on a CDC 3600

Moore's Law in Scientific Computing

Number of particles in MD simulations has doubled:

- **Every 19 months in the past 50 years for classical MD**
- **Every 22 months in the past 30 years for DFT-MD**



10^{12}-atom MD & 10^8-electron DFT on a petaflop/s computer with advances in algorithmic & parallel-computing techniques

ACM Best Dissertation Award

site map accessibility contact

Association for Computing Machinery

Advancing Computing as a Science & Profession

It's algorithm!

SEARCH

join | renew | subscribe

you are here: home → awards → doctoral dissertation award

ACM myACM

- Home
- Membership
- Publications
- Digital Library
- Special Interest Groups (SIGs)
- Online Books & Courses
- Chapters
- Conferences
- Calendar of Events
- Awards
- Educational Activities
- Public Policy
- Online Communities
- Computing News
- Buy Proceedings & Videos

Awards

Home | ACM Awards | Nominations Process | Advanced Grades of Membership | Guide to Establishing an Award | Awards Committees | SIG Awards |

1987 - Leslie Greengard

Series Winner *(1987)*

Citation

For his dissertation "The Rapid Evaluation of Potential Fields in Particle Systems." $O(N^2) \rightarrow O(N)$

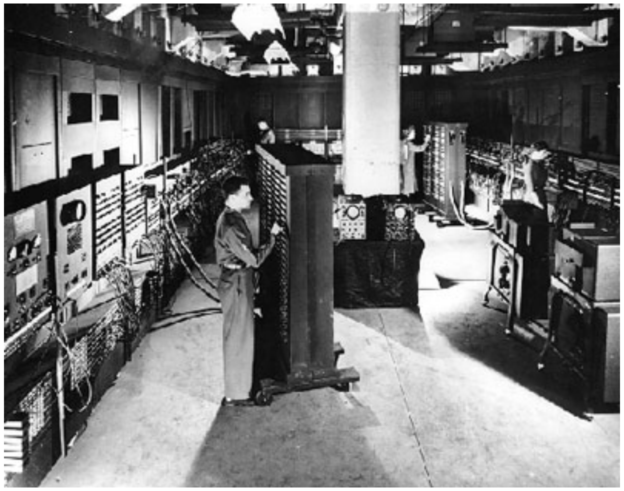
Top 10 Algorithms in History

- Metropolis Algorithm for Monte Carlo
- Simplex Method for Linear Programming
- Krylov Subspace Iteration Methods
- The Decompositional Approach to Matrix Computations
- The Fortran Optimizing Compiler
- QR Algorithm for Computing Eigenvalues
- Quicksort Algorithm for Sorting
- Fast Fourier Transform
- Integer Relation Detection
- Fast Multipole Method

IEEE CiSE, Jan/Feb ('00)

http://awards.acm.org/doctoral%5Fdissertation

History of Supercomputers

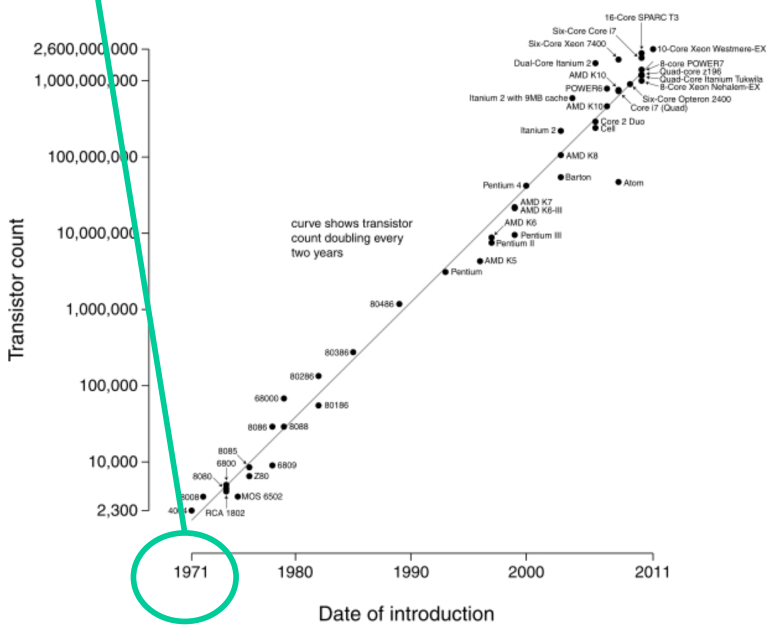


Early ’40s: ENIAC by Presper Eckert & John Mauchly at Univ. of Pennsylvania—first general-purpose electronic computer

’76: Cray 1 by Seymour Cray—beginning of vector supercomputer era

Late 80’s: massively parallel computers such as the Thinking Machines CM-2

(’71: Intel 4004—invention of microprocessor)



ACM Best Dissertation Award

site map accessibility contact

SEARCH

join | renew | subscribe

Association for Computing Machinery

Advancing Computing as a Science & Profession

you are here: home → awards → doctoral dissertation award

ACM myACM

- Home
- Membership
- Publications
- Digital Library
- Special Interest Groups (SIGs)
- Online Books & Courses
- Chapters
- Conferences
- Calendar of Events
- Awards
- Educational Activities
- Public Policy
- Online Communities
- Computing News
- Buy Proceedings & Videos

Awards

Home | ACM Awards | Nominations Process | Advanced Grades of Membership | Guide to Establishing an Award | Awards Committees | SIG Awards |

1985 - Daniel Hillis

Series Winner *(1985)*

Citation

For his dissertation "The Connection Machine."

New Cancer Research Center to be Based at USC

By James Grant on October 26, 2009 7:55 AM

USC Viterbi School of Engineering professor and principal investigator W. Daniel Hillis

E-MAIL PRINT SHARE

USC has been selected to establish a $16 million cancer research center as part of a new strategy against the disease by the U.S. National Institutes of Health and its National Cancer Institute.

The five-year award will create a National Cancer Institute Physical Science-Oncology Center based at USC and involving a consortium of universities. Partnering in the USC grant will be Arizona State University, the California Institute of Technology, Cold Spring Harbor Laboratory, New York University, Stanford University, the University of Arizona and the University of Texas at Austin.

The Physical Science-Oncology Center initiative differs from past cancer research programs. While cancer biologists often work with scientists in other fields, this marks the first large-scale recruitment of outside scientists in the battle against the disease.

Digress: Birth of FORTRAN Programming

Hidden Figures (2016)

N.S. Scott *et al*., *Comput. Phys. Commun*. **252**, 107269 (2020)

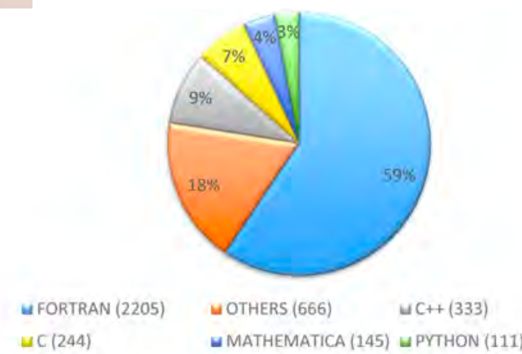


Fig. 13. The range of programming languages used across the Program Library's 3224 published programs (1969–2016).

Math *vs.* Computing

Merge of PC & Supercomputers

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442,010.0	537,212.0	29,899
2	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
3	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
4	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
5	**Perlmutter** - HPE Cray EX235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, HPE DOE/SC/LBNL/NERSC United States	706,304	64,590.0	89,794.5	2,528

Theoretical performance

Measured performance (in Tflop/s)

Flop/s = floating-point operations/second

M (mega) = 10^6
G (giga) = 10^9
T (Tera) = 10^{12}
P (Peta) = 10^{15}
X (Exa) = 10^{18}

http://www.top500.org (June '21)

- USC-CARC: 13,440 cores, 0.62 Pflop/s
- CACS: 4,096 cores
- CACS-INCITE: 200M core-hours/year on 280,320-core 11.7 Pflop/s Theta at Argonne Nat'l Lab

442 petaflop/s Fugaku

Supercomputing Tomorrow

Nation's first exaflop/s computer, Aurora A21

http://aurora.alcf.anl.gov

Delivery

CY 2021

Compute Node

2 Intel Xeon scalable "Sapphire Rapids" processors; 6 X^e arch-based GPUs; Unified Memory Architecture; 8 fabric endpoints

GPU Architecture

X^e arch-based "Ponte Vecchio" GPUTile-based, chiplets, HBM stack, Foveros 3D integration, 7nm

On-Node Interconnect

CPU-GPU: PCIe
GPU-GPU: X^e Link

Aggregate System Memory

> 10 PB

System Interconnect

Cray Slingshot; Dragonfly topology with adaptive routing

Sustained Performance

≥ 1EF DP

High-End Computing at CACS

- **Won two DOE supercomputing awards to develop & deploy metascalable ("design once, scale on future platforms") simulation algorithms (2017-2020)**

U.S. DEPARTMENT OF ENERGY
INCITE
LEADERSHIP COMPUTING

- **NAQMD & RMD simulations on full 800K cores**

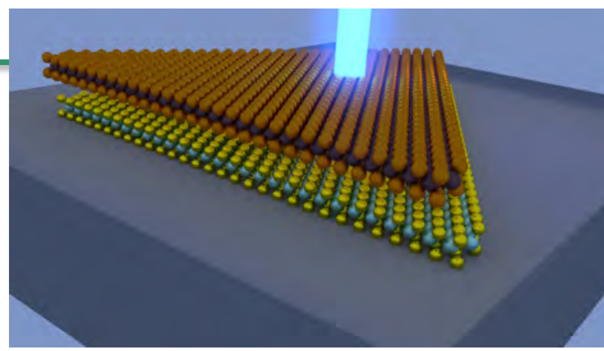
786,432-core IBM Blue Gene/Q

Innovative & Novel Computational Impact on Theory & Experiment

Title: "Petascale Simulations for Layered Materials Genome"

Principal Investigator: Aiichiro Nakano, University of Southern California
Co-Investigator: Priya Vashishta, University of Southern California

AURORA | EARLY SCIENCE PROGRAM



Early Science Projects for Aurora Supercomputer Announced

Metascalable layered materials genome

Investigator: Aiichiro Nakano, University of Southern California

Nation's first exaflop/s computer, Intel A21 (2021)

- **One of 10 initial simulation users of the next-generation DOE supercomputer**

Exaflop/s = 10^{18} floating-point operations per second

Your Platform: USC-CARC

Center for Advanced Research Computing (CARC)

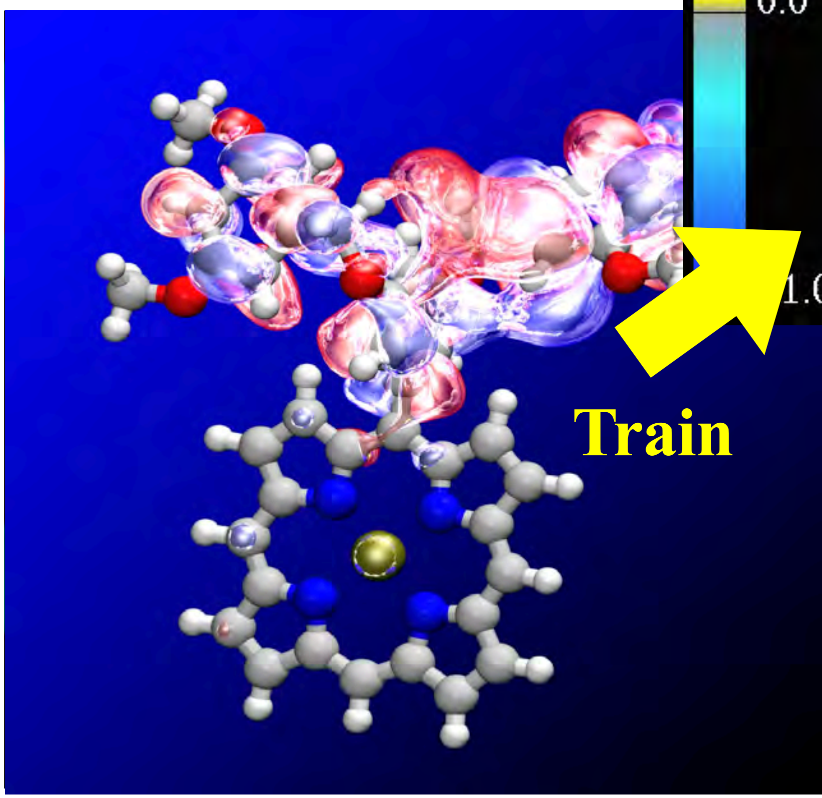
- **The CARC computing resource consists of 2 shared head nodes & a total of 30,000 CPU cores in 1,600 compute nodes. The typical compute node has dual 8 to 16 core processors accelerated by dual GPUs, residing on a 56 gigabit FDR InfiniBand backbone.**
- **Discovery cluster (new):**
  discovery.usc.edu
  discovery2.usc.edu

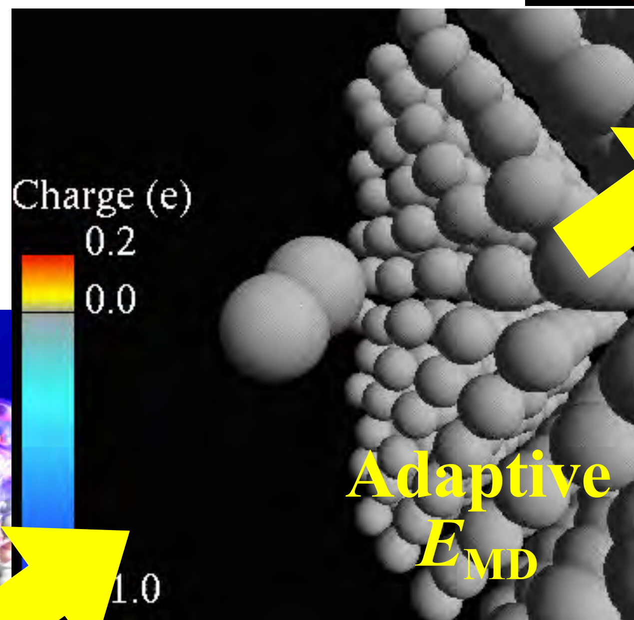
Simulation Engines: NAQMD & RMD

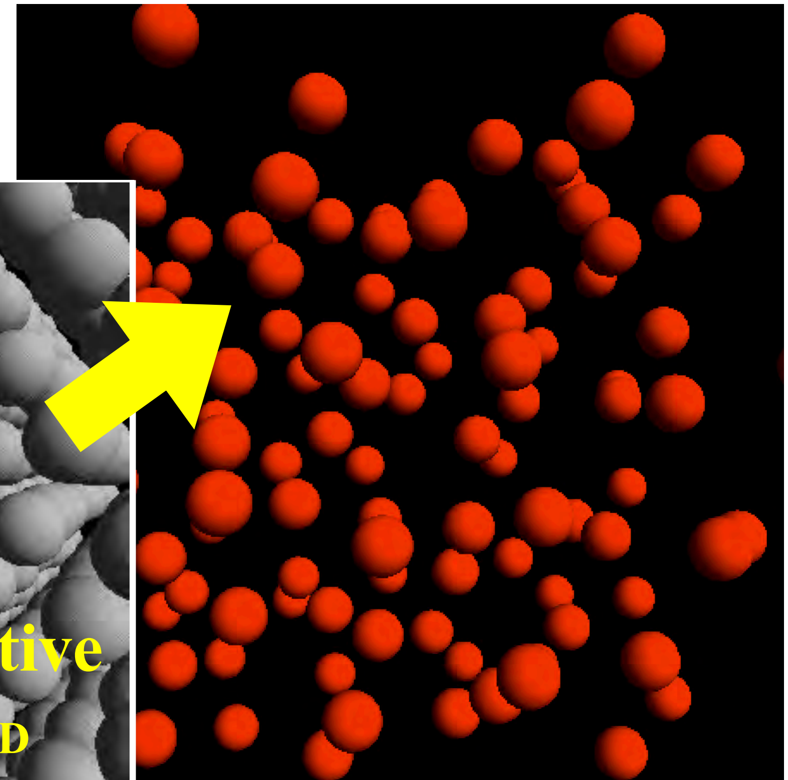
Molecular Dynamics (MD)

Reactive MD (RMD)

Nonadiabatic quantum MD (NAQMD)





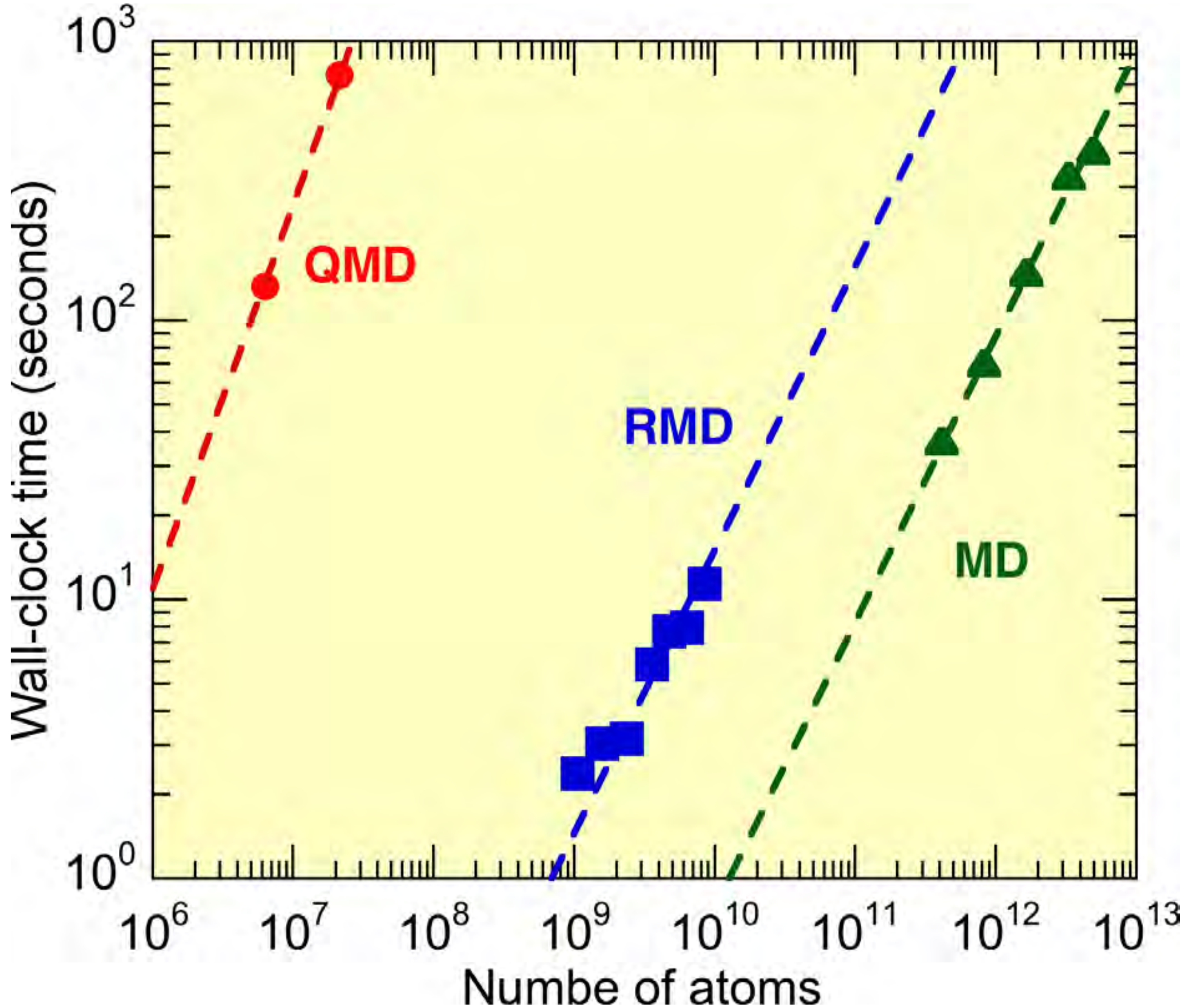


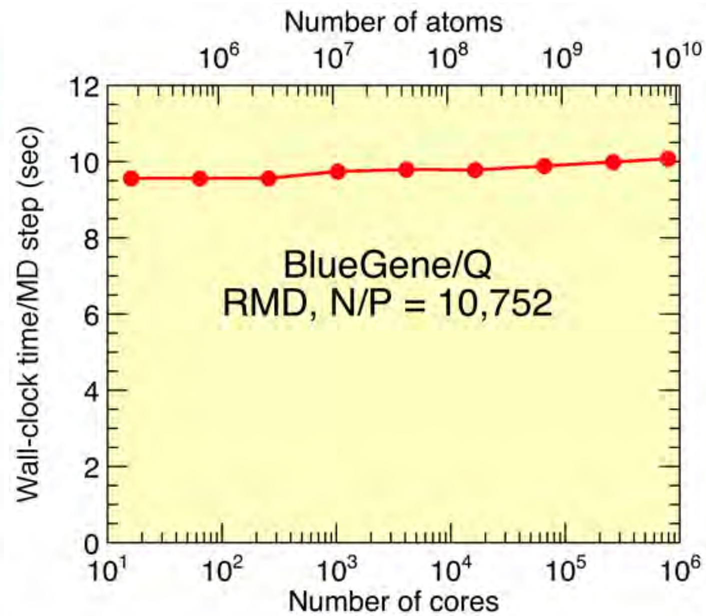
First principles-based reactive force-fields

- **Reactive bond order $\{BO_{ij}\}$**
  → **Bond breakage & formation**
- **Charge equilibration (QEq) $\{q_i\}$**
  → **Charge transfer**

Tersoff, Brenner, Sinnott *et al.*; Streitz & Mintmire *et al.*; van Duin & Goddard (ReaxFF)

Scalable Simulation Algorithm Suite





QMD (quantum molecular dynamics): DC-DFT

RMD (reactive molecular dynamics): F-ReaxFF

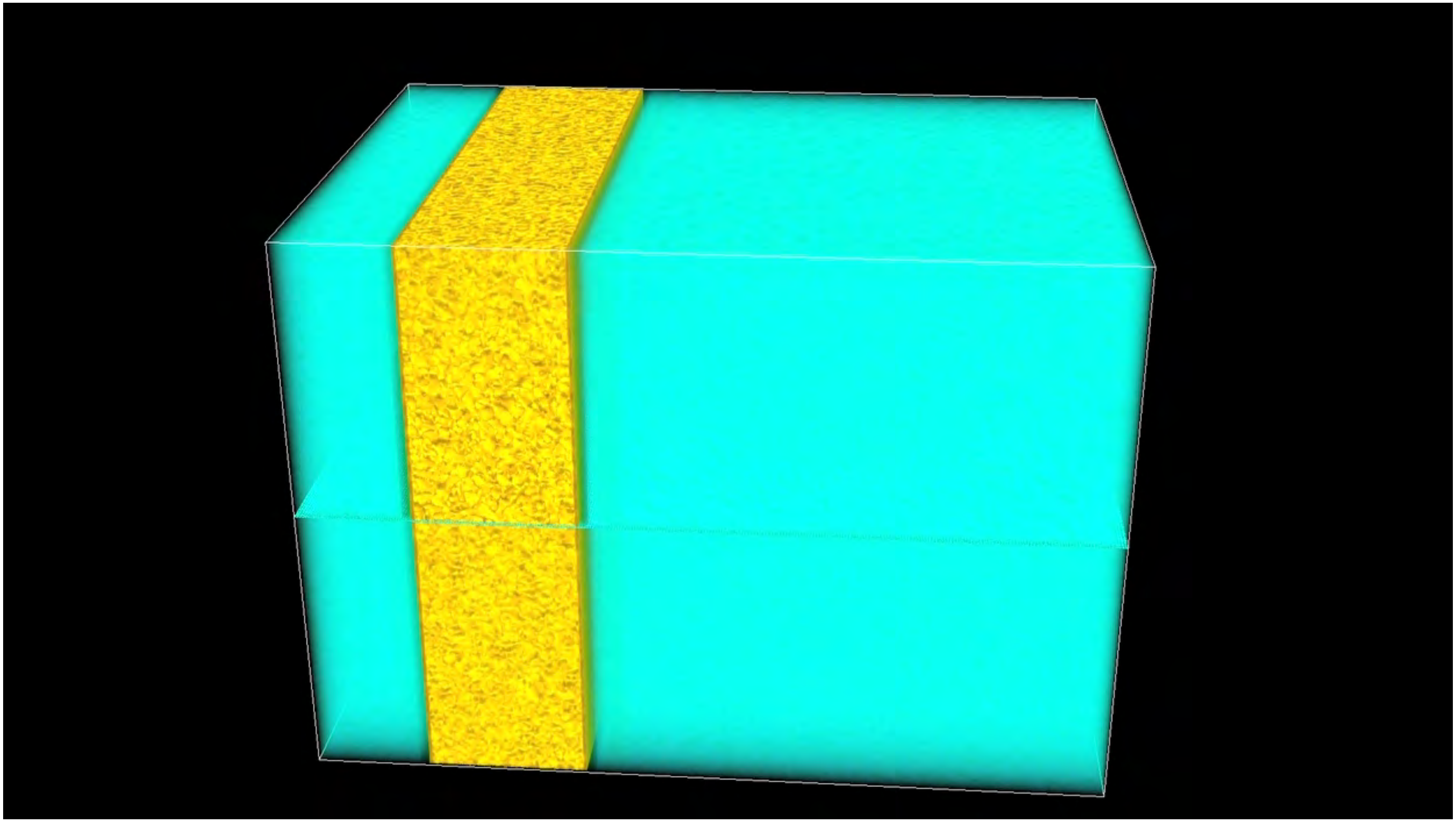
MD (molecular dynamics): MRMD

- **4.9 trillion-atom space-time multiresolution MD (MRMD) of SiO_2**
- **8.5 billion-atom fast reactive force-field (F-ReaxFF) RMD of RDX**
- **39.8 trillion grid points (50.3 million-atom) DC-DFT QMD of SiC**

parallel efficiency over 0.98 on 786,432 Blue Gene/Q cores

Nanobubble Collapse Near Silica Surface

- **Billion-atom MD simulation of shock-induced nanobubble collapse in water near silica surface (67 million core-hours on 163,840 Blue Gene/P cores)**

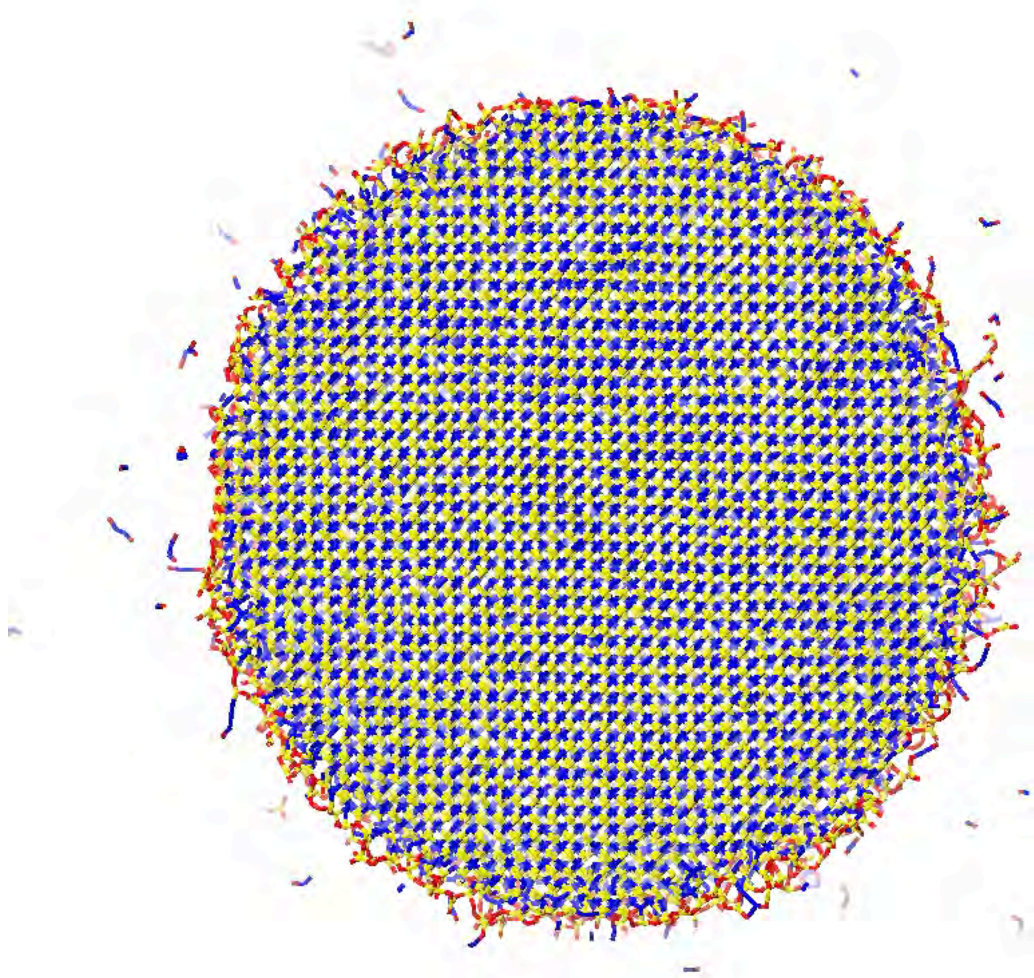


- **Water nanojet formation and its collision with silica surface**

A. Shekhar *et al*., *Phys. Rev. Lett*. **111**, 184503 ('13)

Novel Nano-carbon Synthesis

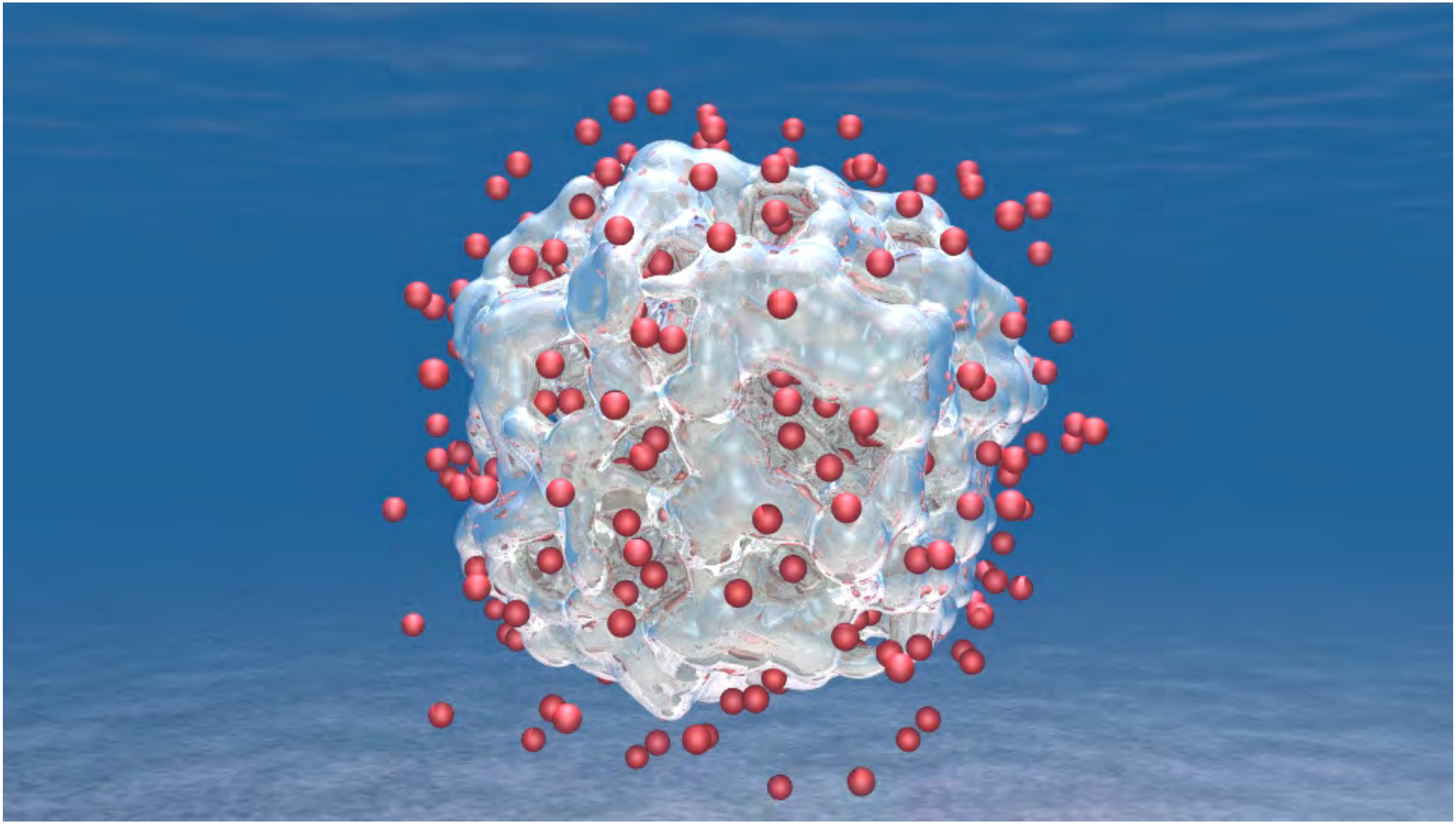
112 million-atom reactive molecular dynamics (RMD) simulation of high-temperature oxidation of SiC nanoparticle on 786,432 IBM Blue Gene/Q cores



K. Nomura *et al.*, *Sci. Rep.* **6**, 24109 ('16)

H_2 Production from Water Using LiAl Particles

16,661-atom quantum molecular dynamics (QMD) simulation of $Li_{441}Al_{441}$ in water on 786,432 IBM Blue Gene/Q cores



K. Shimamura *et al*., *Nano Lett*. **14**, 4090 ('14); K. Nomura *et al*., *SC14* ('14)

Enabling Computer Science Technologies

What can computer science do to enable extreme-scale computational science?

A lot! That's what you will learn in CSCI 596:

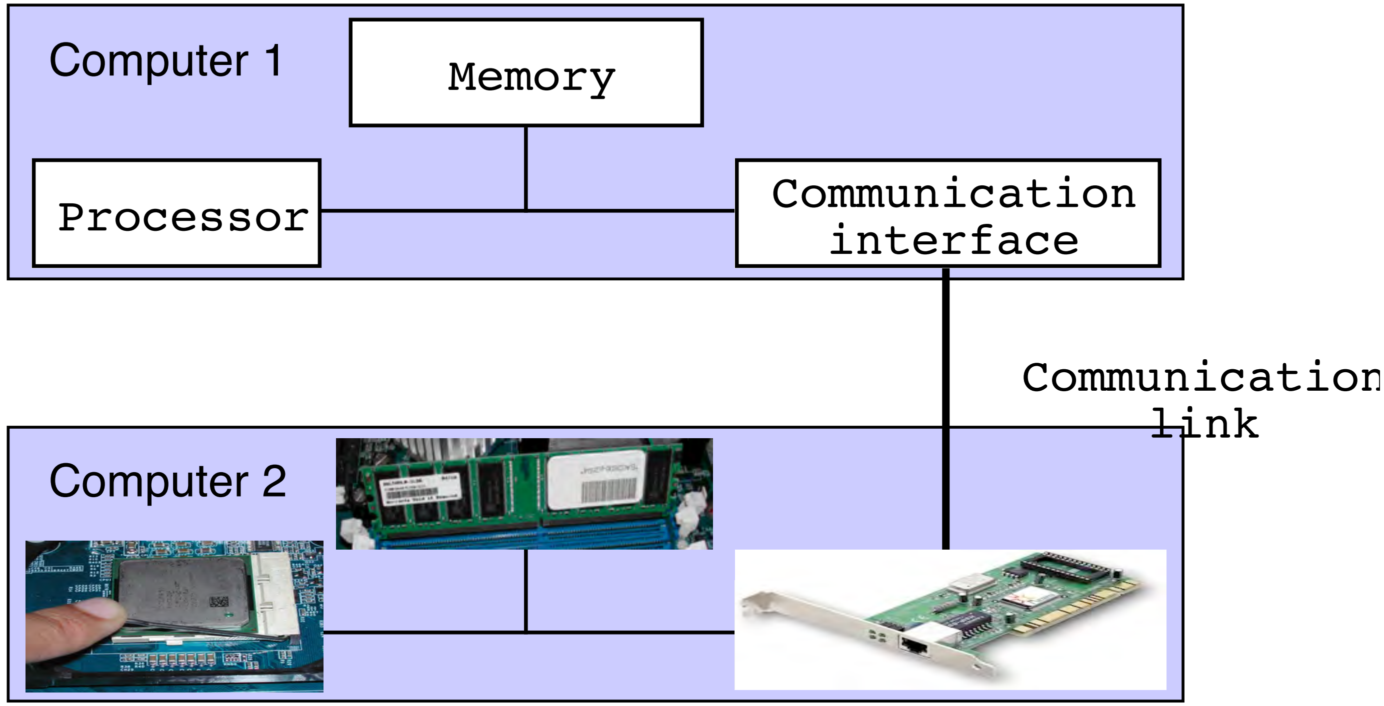
- **Parallel computing**
- **Visualization**
- **Grid/cloud computing**

Parallel Computing

Parallel MD algorithm: Easy!

1. $\vec{v}_i(t+\frac{\Delta}{2}) \leftarrow \vec{v}_i(t) + \frac{\Delta}{2}\vec{a}_i(t)$
2. $\vec{r}_i(t+\Delta) \leftarrow \vec{r}_i(t) + \vec{v}_i(t+\frac{\Delta}{2})\Delta$
3. atom_migrate()
4. atom_cache()
5. Compute $\vec{a}_i(t+\Delta)$ as a function of $\{\vec{r}_i(t+\Delta)\}$
6. $\vec{v}_i(t+\Delta) \leftarrow \vec{v}_i(t+\frac{\Delta}{2}) + \frac{\Delta}{2}\vec{a}_i(t+\Delta)$

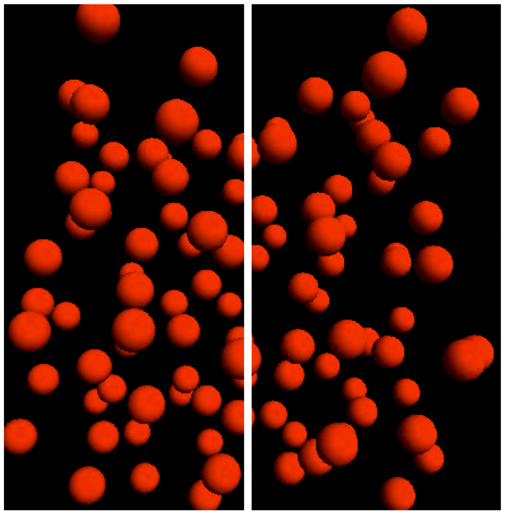
Parallel Computing Hardware

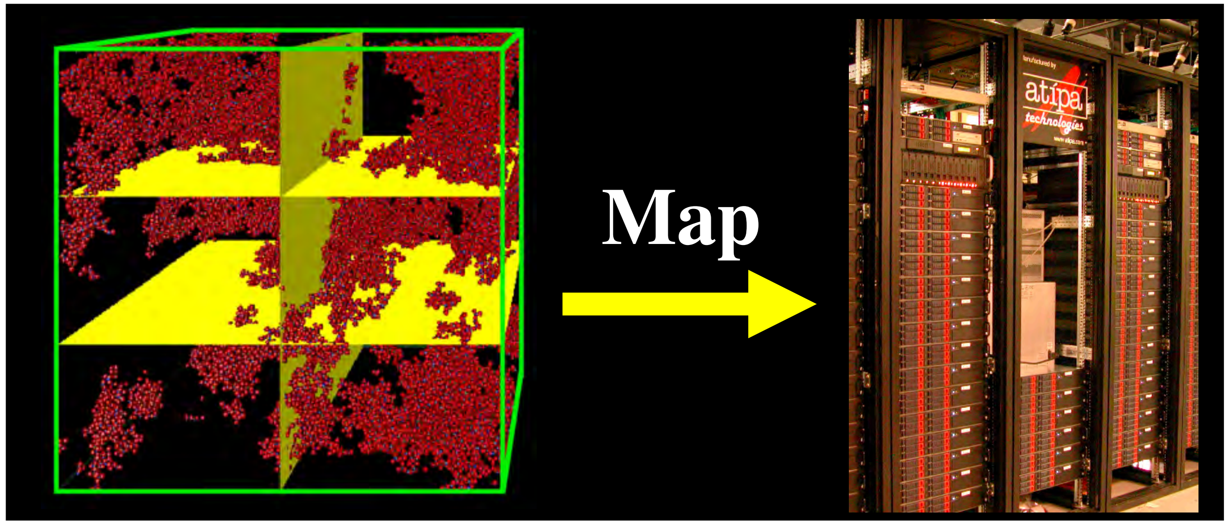


- **Processor:** **Executes arithmetic & logic operations**
- **Memory:** **Stores program & data (stored program computer)**
- **Communication interface:** **Performs signal conversion & synchronization between communication link & a computer**
- **Communication link:** **A wire capable of carrying a sequence of bits as electrical (or optical) signals**

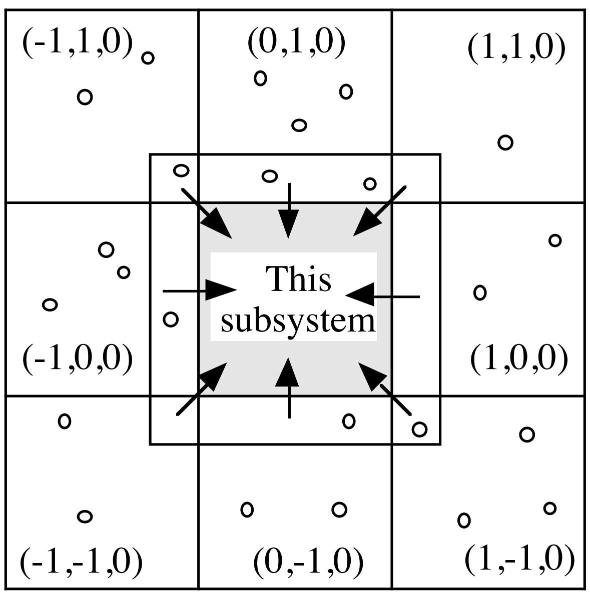
Parallel Molecular Dynamics

Spatial decomposition (short ranged): $O(N/P)$ computation

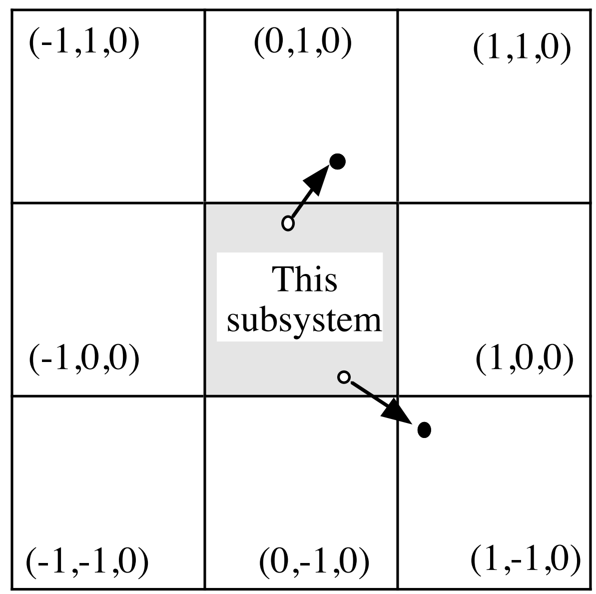




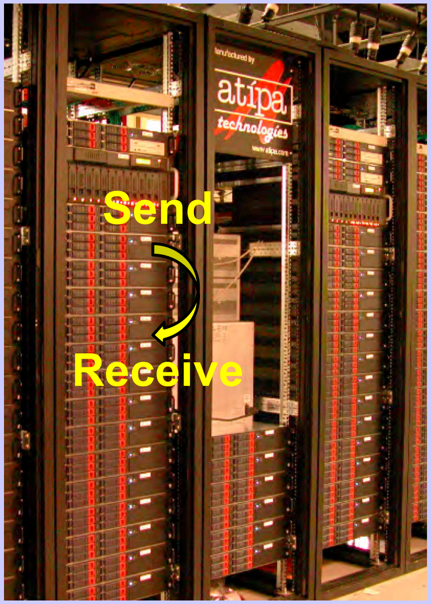
Atom caching: $O((N/P)^{2/3})$



Atom migration



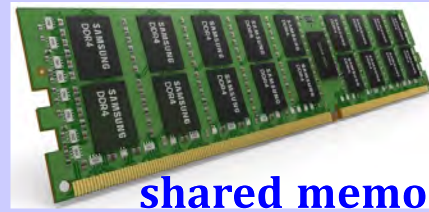
MPI Programming

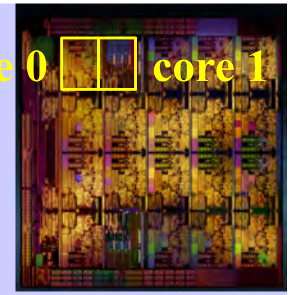



```
#include "mpi.h"
#include <stdio.h>
main(int argc, char *argv[]) {
  MPI_Status status;
  int myid;
  int n;
  MPI_Init(&argc, &argv);
  MPI_Comm_rank(MPI_COMM_WORLD, &myid);
  if (myid == 0) {
    n = 777;
    MPI_Send(&n, 1, MPI_INT, 1, 10, MPI_COMM_WORLD);
  }
  else {
    MPI_Recv(&n, 1, MPI_INT, 0, 10, MPI_COMM_WORLD, &status);
    printf("n = %d\n", n);
  }
  MPI_Finalize();
}
```

http://www.mcs.anl.gov/mpi

OpenMP Programming






```
#include <stdio.h>
#include <omp.h>
void main () {
  int nthreads,tid;
  nthreads = omp_get_num_threads();
  printf("Sequential section: # of threads = %d\n",nthreads);
  /* Fork multi-threads with own copies of variable */
  #pragma omp parallel private(tid)
  {
    /* Obtain & print thread id */
    tid = omp_get_thread_num();
    printf("Parallel section: Hello world from thread %d\n",tid);
    /* Only master thread does this */
    if (tid == 0) {
      nthreads = omp_get_num_threads();
      printf("Parallel section: # of threads = %d\n",nthreads);}
  } /* All created threads terminate */
}
```

parallel section

- **Obtain the number of threads & my thread ID**
- **By default, all variables are shared unless selectively changing storage attributes using private clauses**

http://www.openmp.org

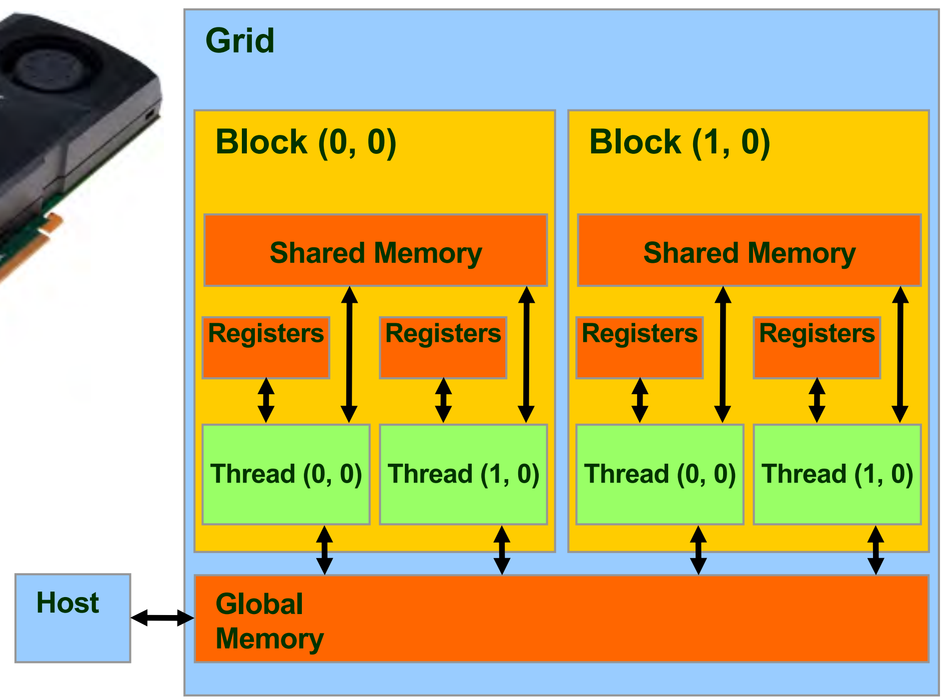
GPU Programming: CUDA

- Compute Unified Device Architecture
- Integrated host (CPU) + device (GPU) application programming interface based on C language developed at NVIDIA
- CUDA homepage
  `http://www.nvidia.com/object/cuda_home_new.html`
- Compilation

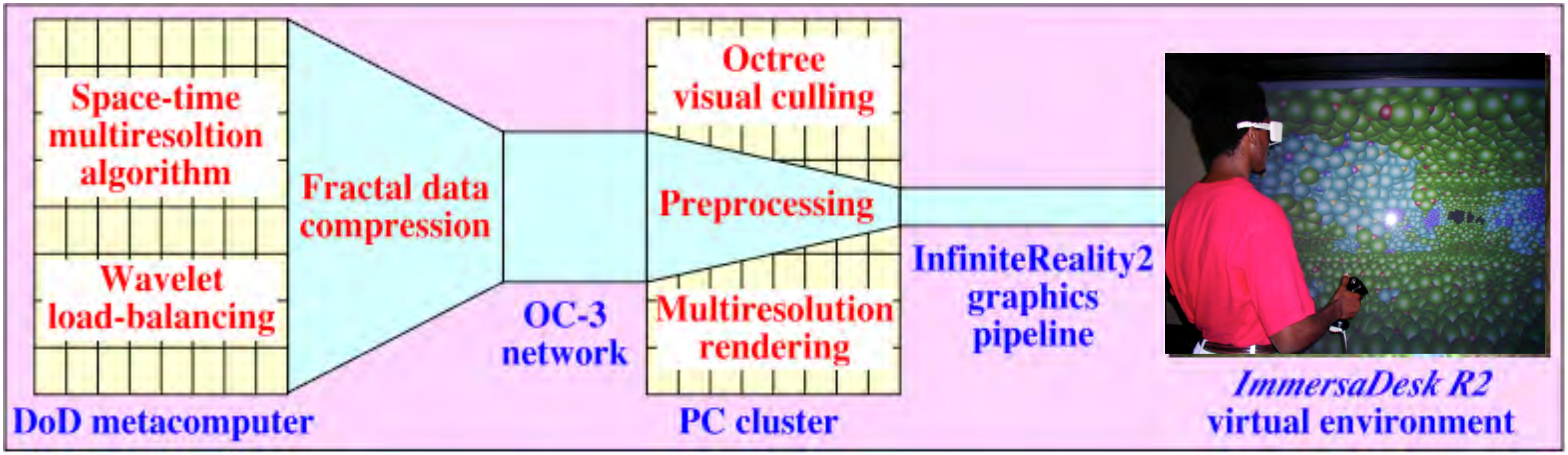
```
$ nvcc pi.cu
```

- Execution

```
$ a.out
PI = 3.141593
```

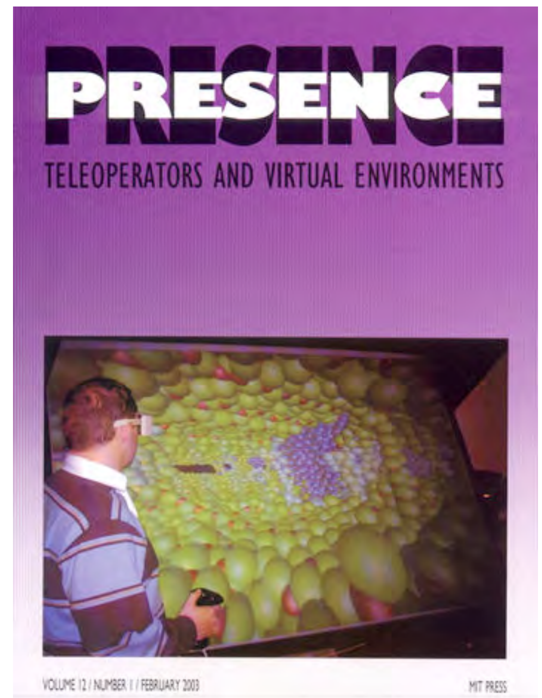


Immersive & Interactive Visualization



Basics:
- OpenGL programming/CAVE library

Billion-atom walkthrough:
- Octree-based view frustum culling
- Probabilistic occlusion culling
- Multiresolution rendering
- Parallel & distributed processing

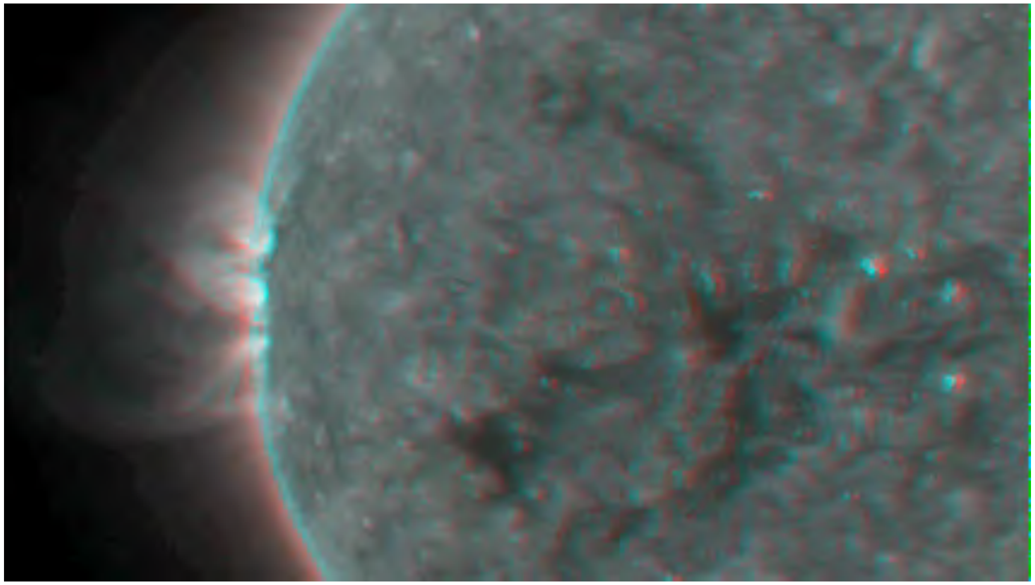


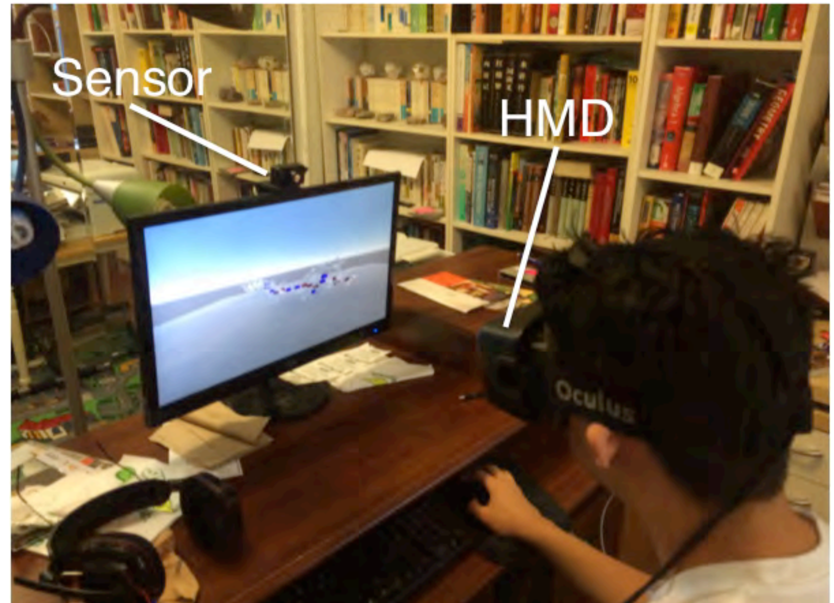
OpenGL Programming

```
...
#include <OpenGL/gl.h>
#include <OpenGL/glu.h>
#include <GLUT/glut.h>
...
/* Set a glut callback function */
glutDisplayFunc(display);

/* generate an OpenGL display list for single sphere */
sphereid = glGenLists(1);
makeFastNiceSphere(sphereid,atom_radius);

/* generate an OpenGL display list for the atoms' geometry */
atomsid = glGenLists(1);
/* make the geometry of the current frame's atoms */
makeCurframeGeom();

/* Start main display loop */
glutMainLoop();
```

http://www.opengl.org

3D Is Back in Hollywood (and Home)

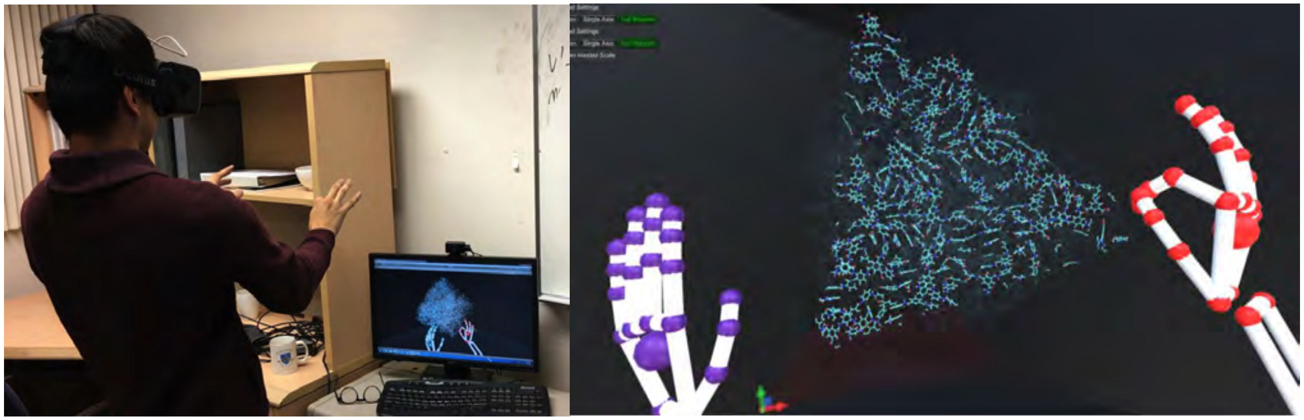




https://github.com/USCCACS/GEARS

Oculus Rift & HTC Vive head mounted displays

Scientific Mixed Reality



VR

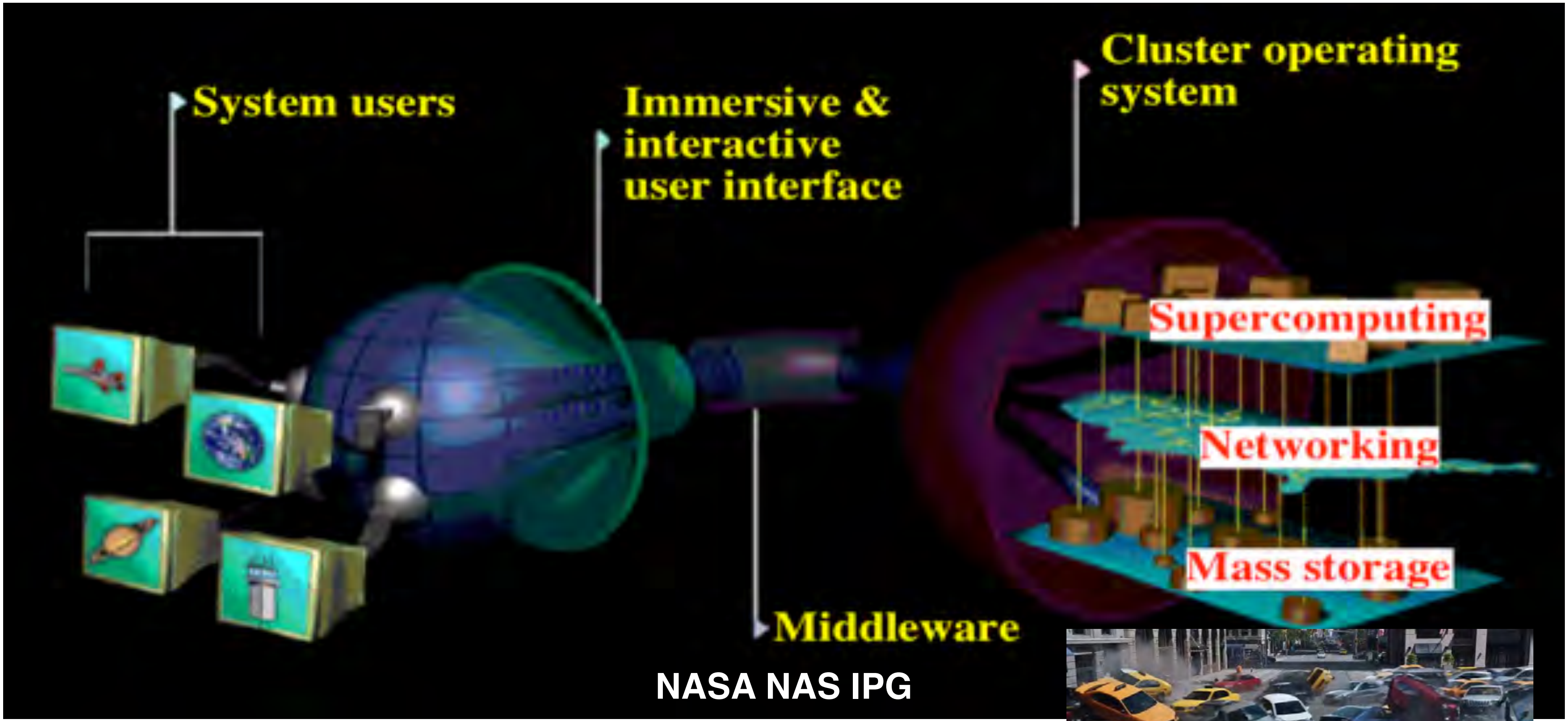
MR

Microsoft Hololens academic seeding program at USC

cf. CSCI 538: Augmented, Virtual and Mixed Reality

Grid Computing

- **World Wide Web:** **Universal interface to digital library on the Internet**
- **Information Grid:** **Pervasive (from any place in the world at any time) access to everything (computing, mass storage, experimental equipments, distributed sensors, *etc*., on high-speed networks)**



Grid Computing

- **Globus middleware**
  - > **Resource monitoring, discovery, & management**
  - > **Security**
  - > **...**

  `http://www.globus.org`

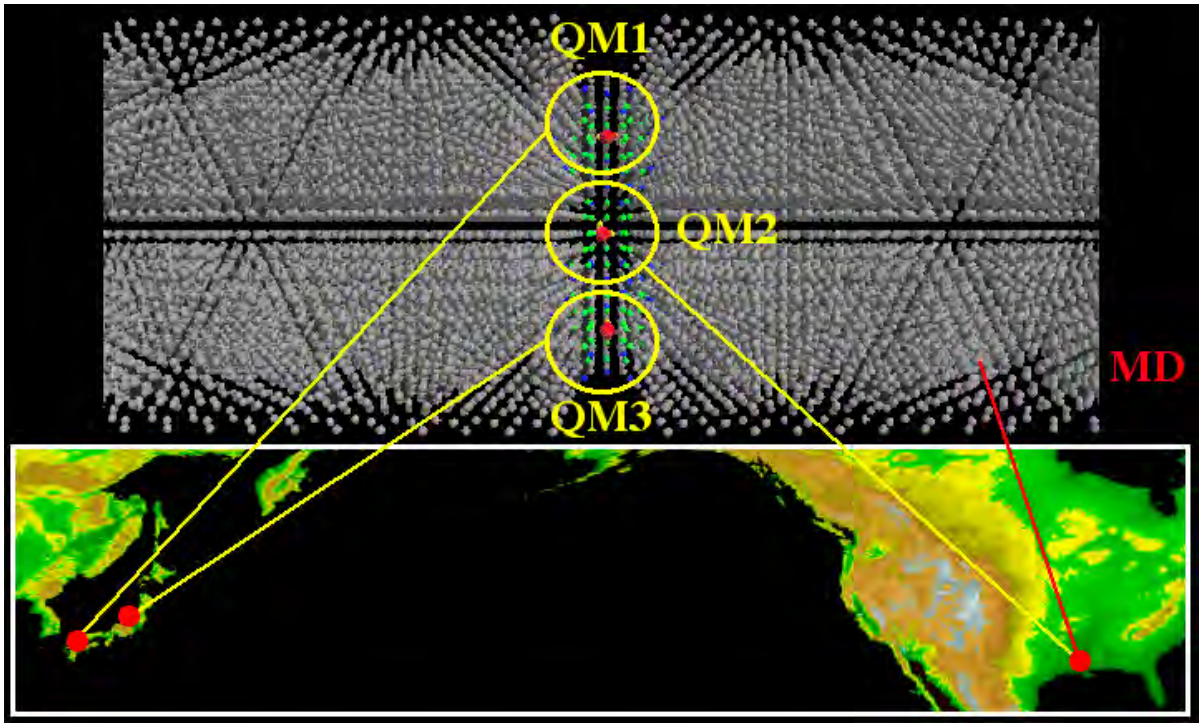
- **Globus-enabled MPI**

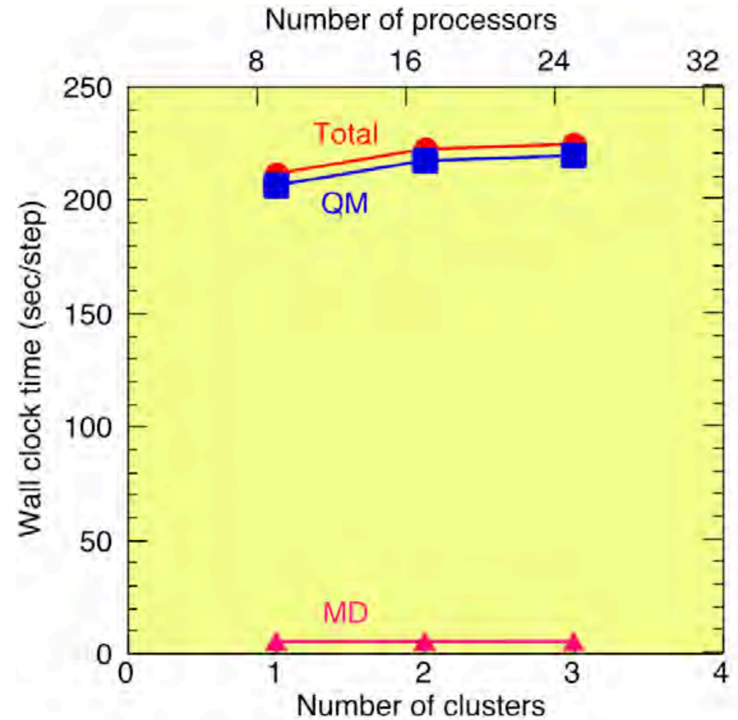
  `http://www.globus.org/grid_software/computation/mpich-g2.php`

Global Collaborative Simulation

Multiscale MD/QM simulation on a Grid of distributed PC clusters in the US & Japan

- Task decomposition (MPI Communicator) + spatial decomposition
- MPICH-G2/Globus





Japan: Yamaguchi—65 P4 2.0GHz
Hiroshima, Okayama, Niigata—3×24 P4 1.8GHz
US: Louisiana—17 Athlon XP 1900+

MD — 91,256 atoms
QM (DFT) — 76*n* atoms on *n* clusters

- Scaled speedup, $P = 1$ (for MD) + $8n$ (for QM)
- Efficiency = 94.0% on 25 processors over 3 PC clusters

Kikuchi *et al.* *IEEE/ACM SC02*

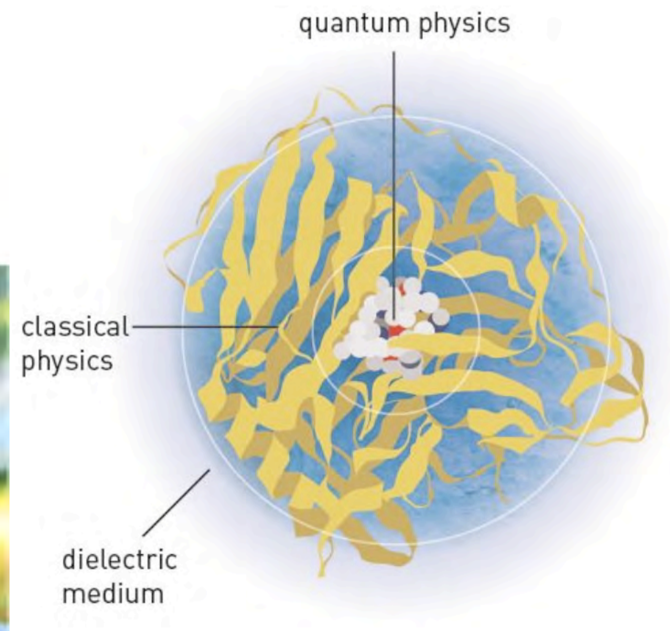
Multiscale Modeling

The Nobel Prize in Chemistry 2013

© Nobel Media AB
Martin Karplus

Photo: Keilana via Wikimedia Commons
Michael Levitt

Photo: Wikimedia Commons
Arieh Warshel



QM/MM:
quantum-mechanical/molecular-mechanical modeling

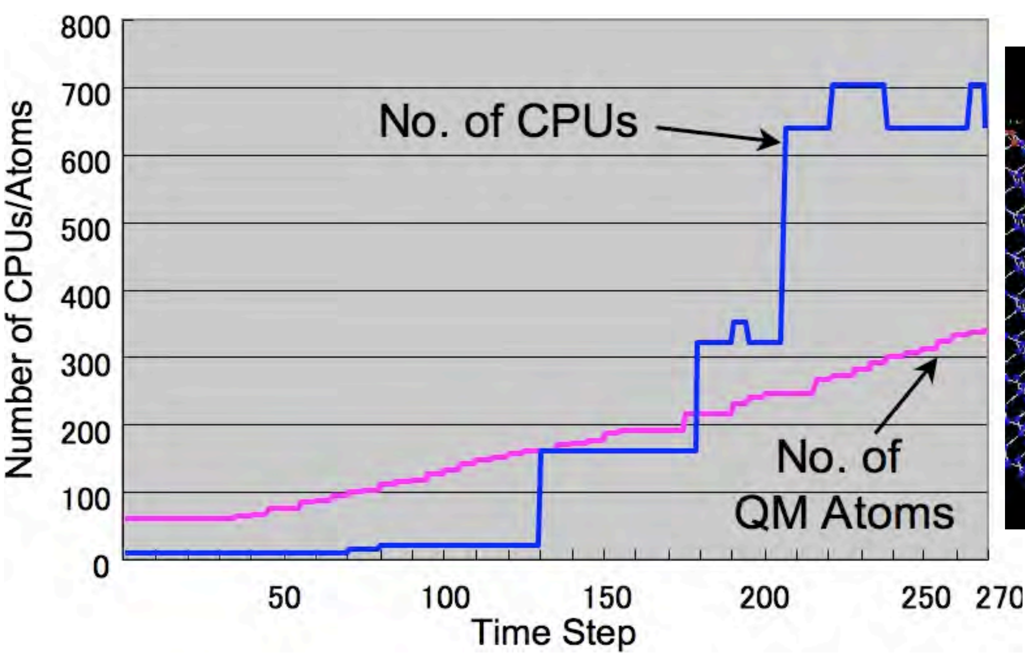
The Nobel Prize in Chemistry 2013 was awarded jointly to Martin Karplus, Michael Levitt and Arieh Warshel *"for the development of multiscale models for complex chemical systems"*.

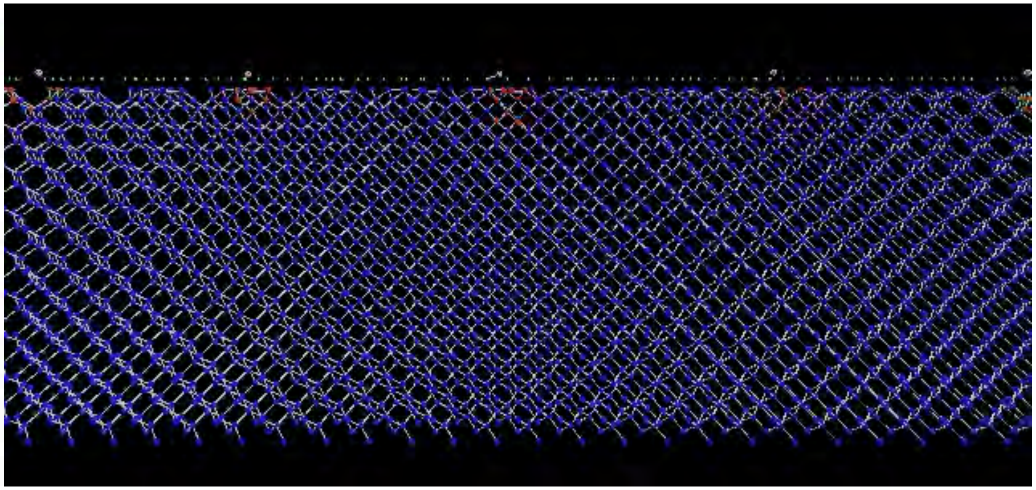
A. Warshel & M. Karplus, *J. Am. Chem. Soc*. **94**, 5612 (’72)
A. Warshel & M. Levitt, *J. Mol. Biol*. **103**, 227 (’76)

Sustainable Grid Supercomputing

- Sustained (> months) supercomputing (> 10^3 CPUs) on a Grid of geographically distributed supercomputers
- Hybrid Grid remote procedure call (GridRPC) + message passing (MPI) programming
- Dynamic allocation of computing resources on demand & automated migration due to reservation schedule & faults

Ninf-G GridRPC: ninf.apgrid.org; MPICH: www.mcs.anl.gov/mpi

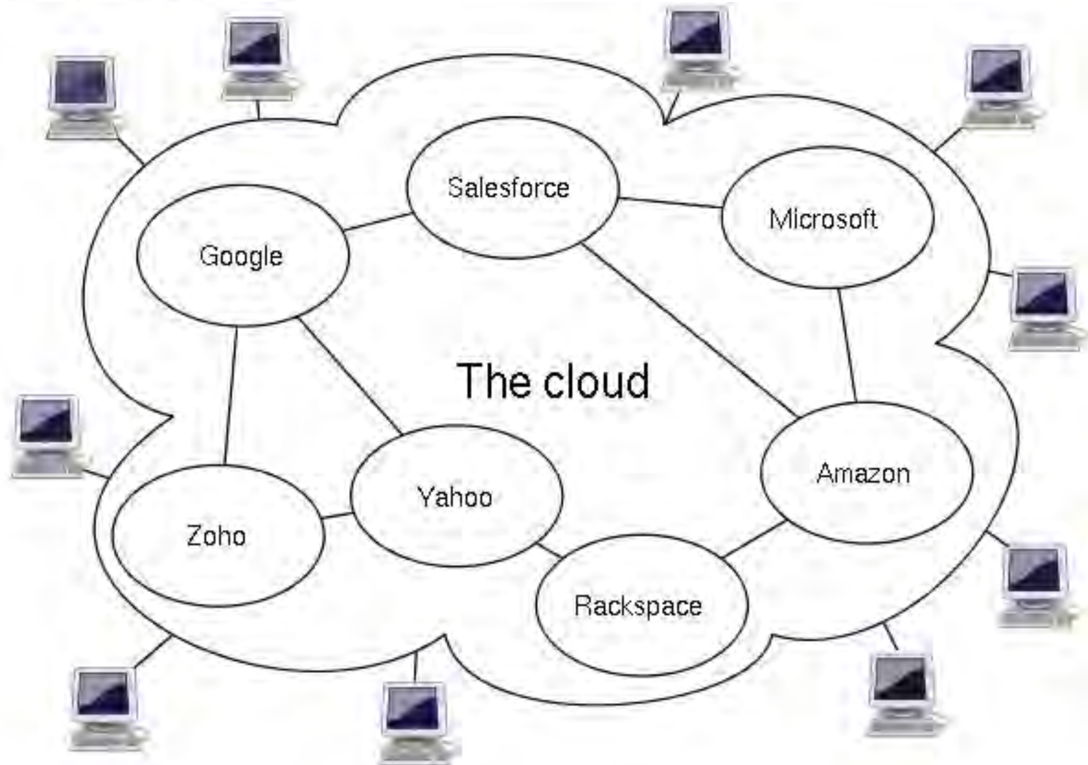




Takemiya *et al*., *IEEE/ACM SC06*
Song *et al*., *IJCS* ('09)

Multiscale QM/MD simulation of high-energy beam oxidation of Si

Above the Clouds: A Berkeley View of Cloud Computing



Michael Armbrust
Armando Fox
Rean Griffith
Anthony D. Joseph
Randy H. Katz
Andrew Konwinski
Gunho Lee
David A. Patterson
Ariel Rabkin
Ion Stoica
Matei Zaharia

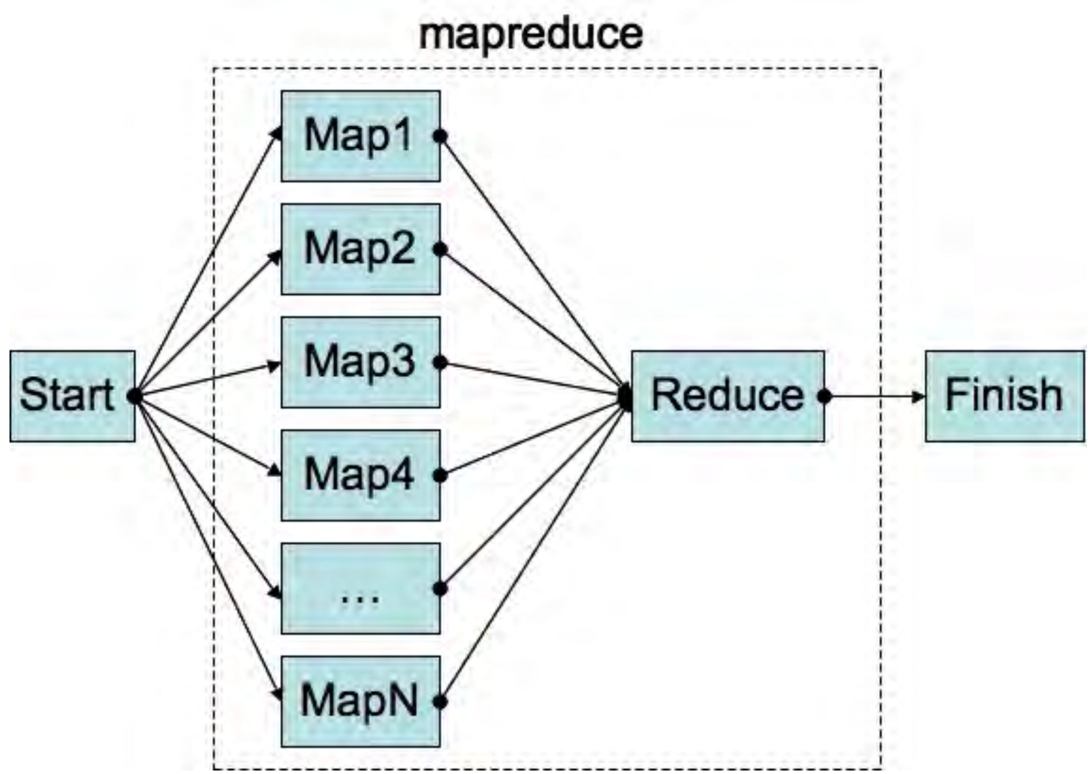
Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2009-28
http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html

February 10, 2009

MapReduce

- Parallel programming model for data-intensive applications on large clusters
  - > User implements Map() and Reduce()
- Parallel computing framework
  - > Libraries take care of everything else
    - Parallelization
    - Fault tolerance
    - Data distribution
    - Load balancing
- Developed at Google

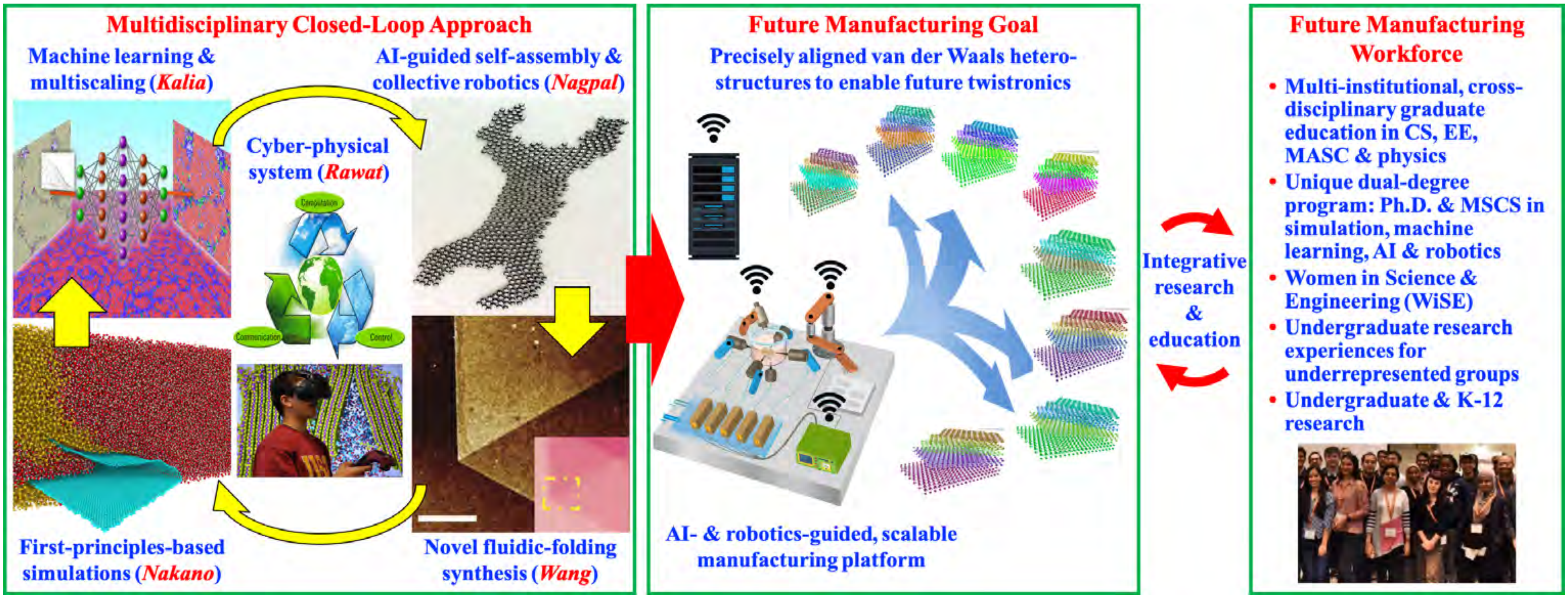


Cybermanufacturing

NSF 2036359 FMRG: Artificial Intelligence Driven Cybermanufacturing of Quantum Material Architectures

9/1/2020 – 8/31/2025

R. Nagpal (*Harvard*); R. Kalia, A. Nakano, H. Wang (*USC*); D. Rawat (*Howard*)



This project develops a transformative future manufacturing platform for quantum material architectures using a cybermanufacturing approach, which combines artificial intelligence, robotics, multiscale modeling, and predictive simulation for the automated & parallel assembly of multiple two-dimensional materials into complex three-dimensional structures.

Computational Research Survival Guide

Talk to user
Regular expression (text processing)

https://regexone.com

CSCI 596: Talk to (C) compiler via libraries & directives
Parallel processing: MPI, OpenMP, CUDA, ...
Visualization: OpenGL

Talk to operating system
Shell programming

A. Scopatz and K. D. Huff, *Effective Computation in Physics* (O'Reilly, '15)
http://www.amazon.com/Effective-Computation-Physics-Anthony-Scopatz/dp/1491901535