CS410
Parallel Computing:
An Introduction

Milind Chabbi

Nikhil Hegde

Department of Computer Science
IIT Dharwad

Slide adapted from John Mellor-Crummey's Comp422 @ Rice University

Course Information

- **Time: M,W,F**
- **Delivery mode:**
  - **Hybrid:**
    - **Online via google class room (Milind Chabbi)**
    - **In class (Nikhil Hegde)**
- **Instructors:**
  - **Milind Chabbi and Nikhil Hegde**
    - **{chabbi.milind, nikhilh}@iitdh.ac.in**
    - **email through google class room preferred**
- **Teaching Assistants:**
  - ~~**Vivek Share**~~
    - ~~**Vivek Shahare <212011006@iitdh.ac.in>**~~

Parallelism

- **Parallelism: simultaneous execution of multiple parts of a computation**
- **Reasons for applying parallelism**
  - **—speed up: solve a problem faster**
    - **i.e. strong scaling**
  - **—scale up: solve a larger problem**
    - **i.e. weak scaling**
  - **—scale out: solve many problems**
    - **"… screen billions of small molecules against all the major non-structural proteins of SARS-CoV-19"**
- **CS410 focus: speed up - solving problems faster**

Units of Parallelism?

- Terms
  - —thread
    - a sequence of instructions
    - managed by the OS or a runtime system
  - —process
    - address space containing one or more threads
- How big are the units of parallelism?
  - bit-level parallelism: operate on multiple bits in a word
  - vector instructions: operate on multiple independent bytes or words
  - instruction-level parallelism: > 1 instructions from a thread at once
  - multiple threads in a process
  - multiple processes
- COMP 410 focus: thread and process parallelism

Course Objectives

- **Learn fundamentals of parallel computing**
  - —principles of parallel algorithm design
  - —programming models and methods
  - —parallel computer architectures
  - —parallel algorithms
  - —modeling and analysis of parallel programs and systems
- **Develop skill writing parallel programs**
  - —programming assignments
- **Develop skill analyzing parallel computing problems**
  - —solving problems posed in class

Recommended Books

- **Introduction to Parallel Computing, 2nd Ed, Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar (2003)**
- **Using OpenMP: Portable Shared Memory Parallel Programming - Barbara Chapman, Gabriele Jost, Ruud van der Pas (2008) [PDF online]**
- **Using MPI: Portable Parallel Programming with the Message-Passing Interface, 3rd Ed - William Gropp, Ewing Lusk, Anthony Skjellum (2014)**
- **Programming Massively Parallel Processors: A Hands-on Approach, 3rd Ed. - David B. Kirk, Wen-mei W. Hwu (2016) [PDF of 2nd edition available online]**

Topics (Part 1)

- Introduction
- Principles of parallel algorithm design
  - —decomposition techniques
  - —mapping & scheduling computation
  - —templates
- Programming shared-address space systems
  - —Cilk
  - —OpenMP
  - —Pthreads
  - —synchronization
- Parallel computer architectures
  - —shared memory systems and cache coherence
  - —distributed-memory systems
  - —interconnection networks and routing

Topics (Part 2)

- Programming scalable systems
  - —message passing: MPI
  - —global address space languages
- Collective communication
- Analytical modeling of program performance
  - —speedup, efficiency, scalability, cost optimality, isoefficiency
- Parallel algorithms
  - —non-numerical algorithms: sorting, graphs
  - —numerical algorithms: dense and sparse matrix algorithms
- Performance measurement and analysis of parallel programs
- GPU Programming with CUDA

Prerequisites

- **Programming in C, C++, or similar**
- **Basics of data structures**
- **Basics of machine architecture**
- **See the instructor(s) if you have concerns**

Motivations for Parallel Computing

- **Technology push**
- **Application pull**

The Rise of Multicore Processors

Advance of Semiconductors: “Moore’s Law”

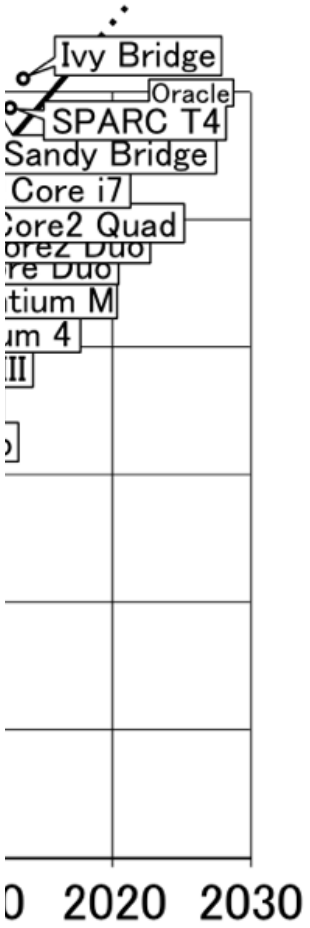
Gordon Moore, Founder of Intel

- **1965: since the integrated circuit was invented, the number of transistors in an integrated circuit has roughly doubled every year; this trend would continue for the foreseeable future**
- **1975: revised - circuit complexity doubles every two years**

Gordon Moore

Moore in 1978

Born	Gordon Earle Moore January 3, 1929 Pescadero, California, U.S.[3]
Died	March 24, 2023 (aged 94) Waimea, Hawaii, U.S.

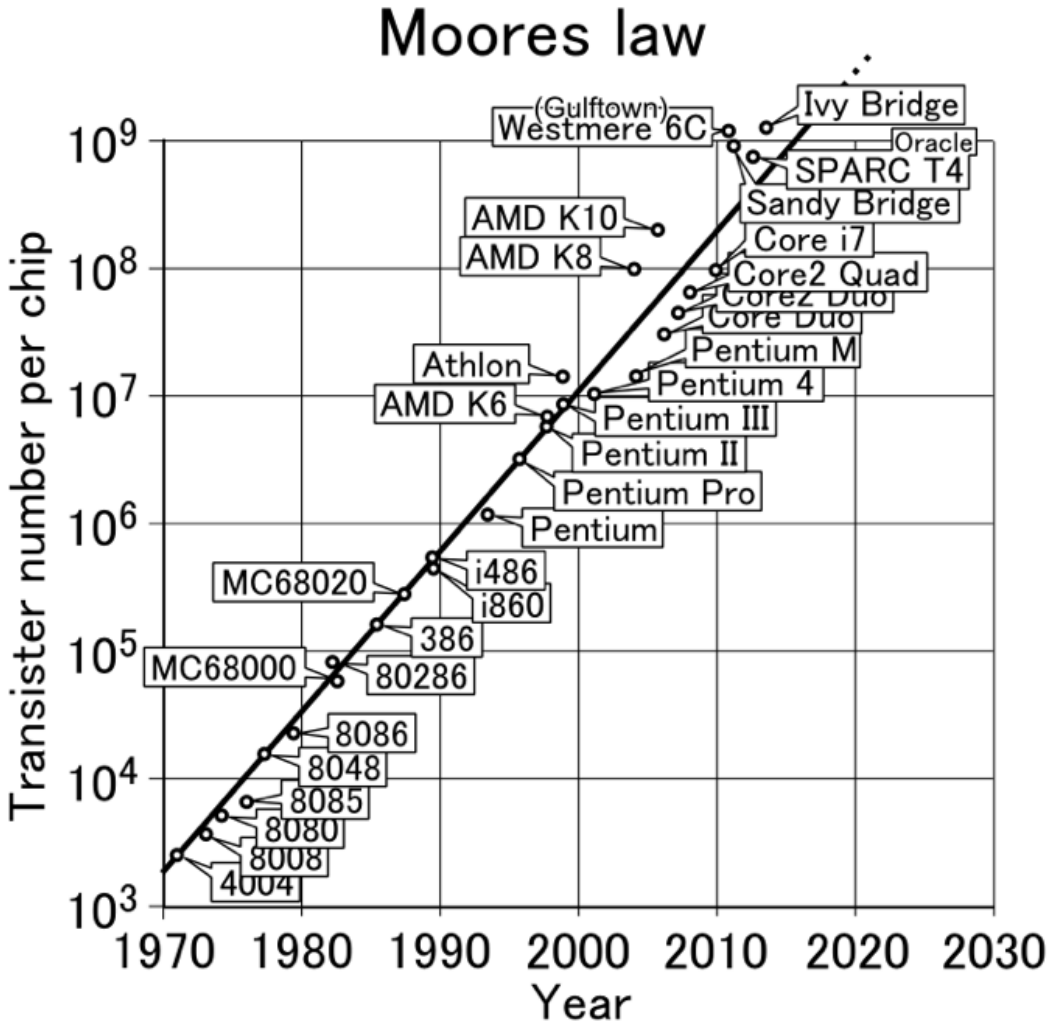


dia Commons

Advance of Semiconductors: “Moore’s Law”

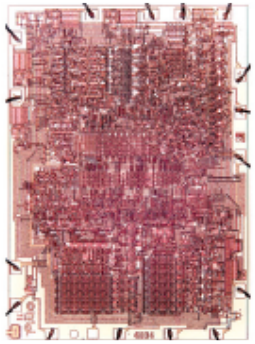
Gordon Moore, Founder of Intel

- **1965: since the integrated circuit was invented, the number of transistors in an integrated circuit has roughly doubled every year; this trend would continue for the foreseeable future**
- **1975: revised - circuit complexity doubles every two years**

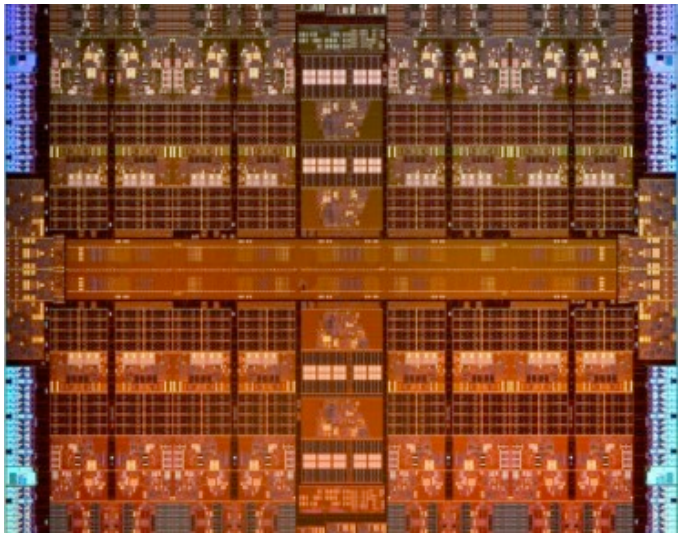


By shigeru23 CC BY-SA 3.0, via Wikimedia Commons

Evolution of Microprocessors 1971-2017

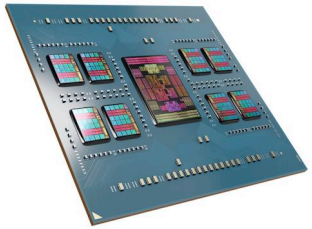


Intel 4004, 1971
1 core, no cache
23K transistors

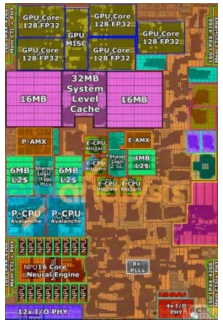


Oracle M7, 2015
32 cores, 64MB cache
10B transistors

Nvidia Ampere A100, 2022
6912 CUDA cores
54B transistors



AMD EPYC Bergamo (4th gen/97X4 series) 9-chip module (up to 128 cores and 256 MB (L3) + 128 MB (L2) cache), 2023
82B transistors



Apple M2 Ultra (two M2 Max dies), 2023
132B transistors

Cerebras Wafer Scale Engine 2

Cerebras WSE-2
The Largest Chip Ever Built

46,225	mm^2 silicon
2.6	Trillion transistors
850,000	AI optimized cores
40	Gigabytes on-chip memory
20	Petabyte/s memory bandwidth
220	Petabit/s fabric bandwidth
7nm	Process technology at TSMC

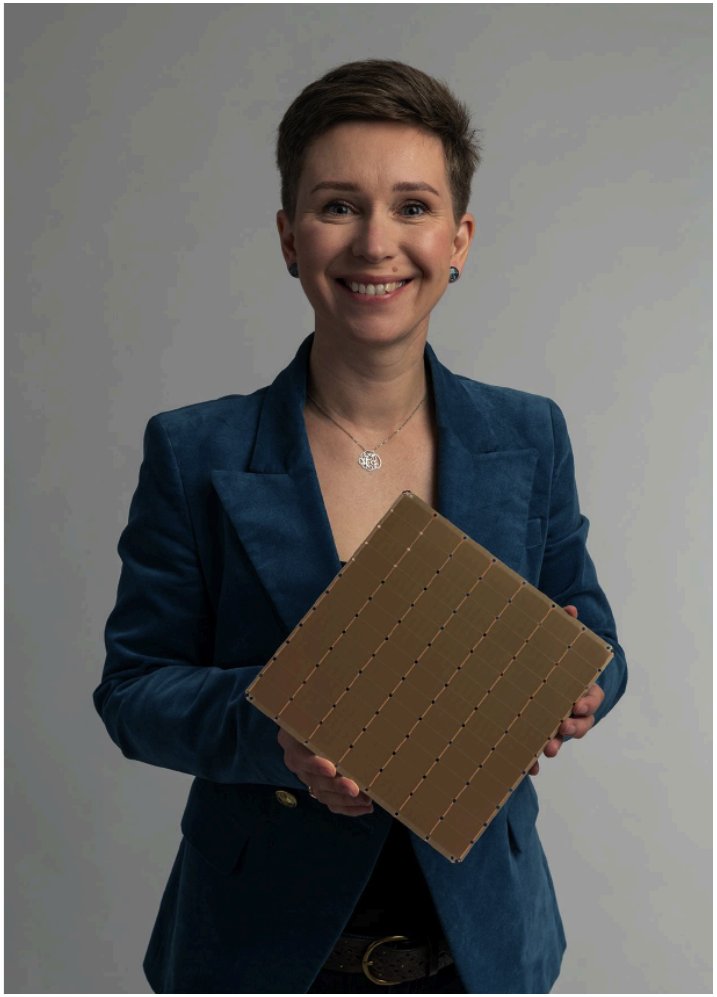


Figure Credit: Sean Lie, Multi-Million Core, Multi-Wafer AI Cluster. Hot Chips 33. August 22-24, 2021. (Virtual Event)

Leveraging Moore's Law Trends

From increasing transistor count to performance

- More transistors = ↑ opportunities for exploiting parallelism
- Parallelism in a CPU core
  - —implicit parallelism: invisible to the programmer
    - pipelined execution of instructions
    - multiple functional units for multiple independent pipelines
  - —explicit parallelism
    - long instruction words (VLIW)
      - bundles of independent instructions that can be issued together
        - e.g., Intel Itanium processor 2000-2017
    - SIMD processor extensions up to 512 bits wide (AVX512)
      - integer, floating point, complex data
      - operations on up to 16 32-bit data items per instruction

Microprocessor Architecture (Mid 90's)

- **Superscalar (SS) designs were the state of the art**
  - —multiple functional units (e.g., int, float, branch, load/store)
  - —multiple instruction issue
  - —dynamic scheduling: HW tracks instruction dependencies
  - —speculative execution: look past predicted branches
  - —non-blocking caches: multiple outstanding memory operations
- **Apparent path to higher performance?**
  - —wider instruction issue
  - —support for more speculation

Trouble on the Horizon

Increasing issue width provides diminishing returns

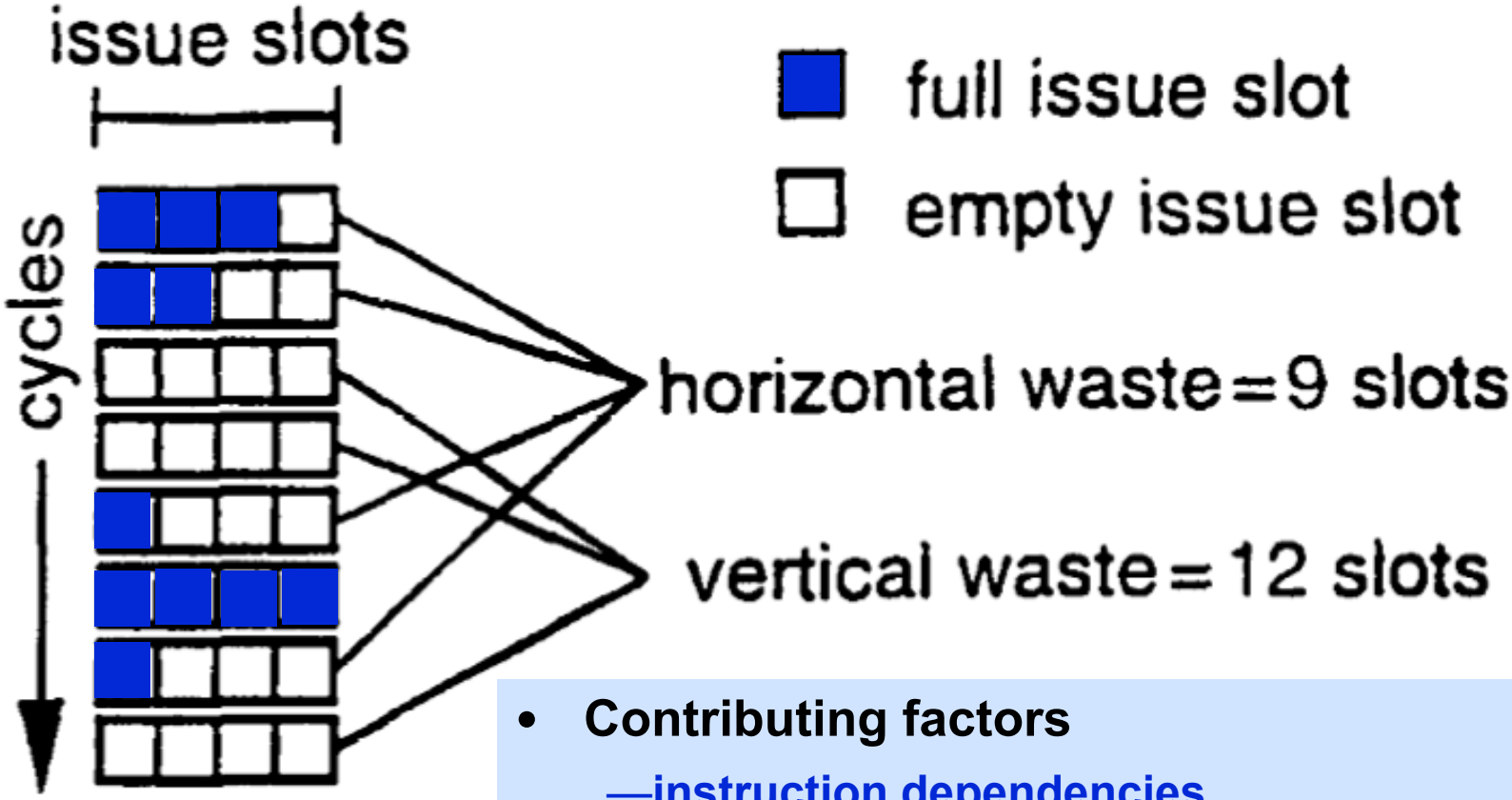
Two factors[1]

- **Fundamental circuit limitations**
  - **—delays ↑ as issue queues ↑ and multi-port register files ↑**
  - **—increasing delays limit performance returns from wider issue**
- **Limited amount of instruction-level parallelism**
  - **—inefficient for programs with difficult-to-predict branches**

[1] The case for a single-chip multiprocessor, K. Olukotun, B. Nayfeh, L. Hammond, K. Wilson, and K. Chang, ASPLOS-VII, 1996.

Instruction-level Parallelism Concerns

Issue Waste



- **Contributing factors**
  - —instruction dependencies
  - —long-latency operations within a thread

Some Sources of Wasted Issue Slots

- TLB miss
- I cache miss
- D cache miss
- Load delays (L1 hits)

Memory Hierarchy

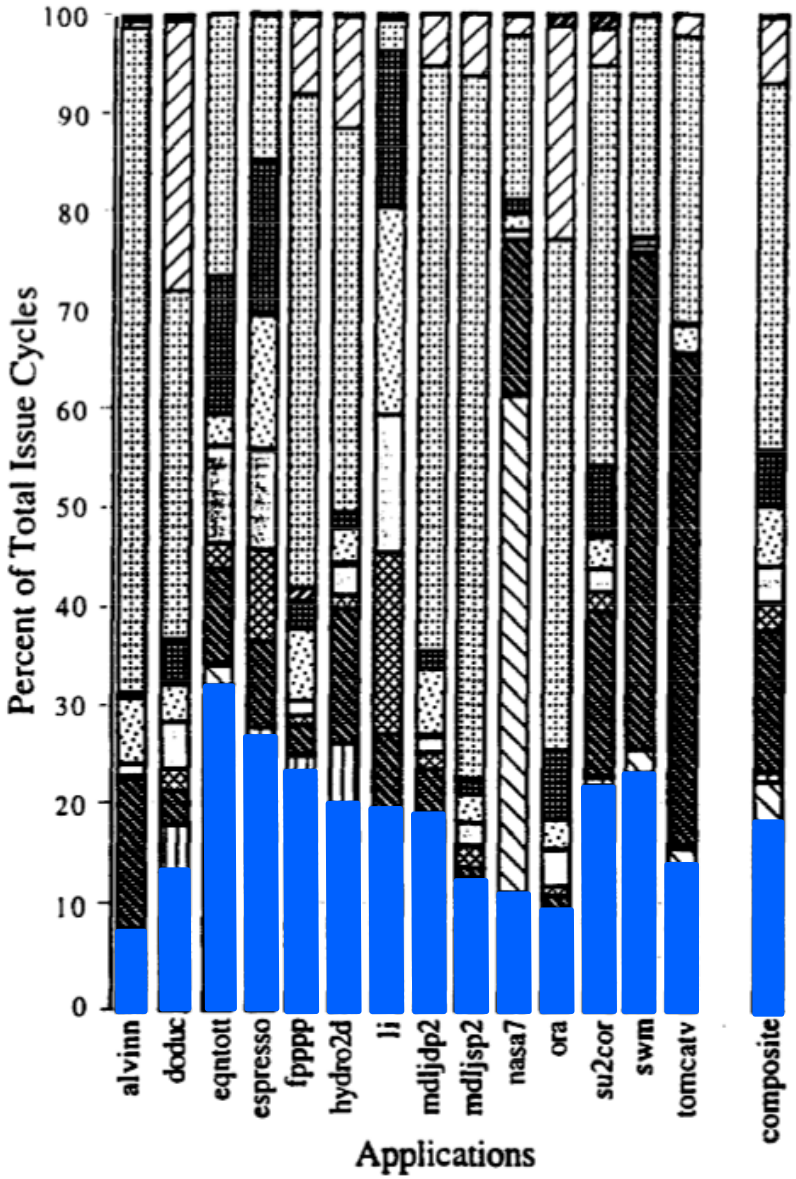
- Branch misprediction

Control Flow

- Instruction dependences
- Memory conflict

Instruction Stream

Simulations of 8-issue Superscalar



Simultaneous multithreading: maximizing on-chip parallelism, Tullsen et. al. ISCA, 1995.

Summary:
Highly underutilized

Applications: most of SPEC92

- On average < 1.5 IPC (19%)
- Dominant waste differs by application
- Short FP dependences: 37%

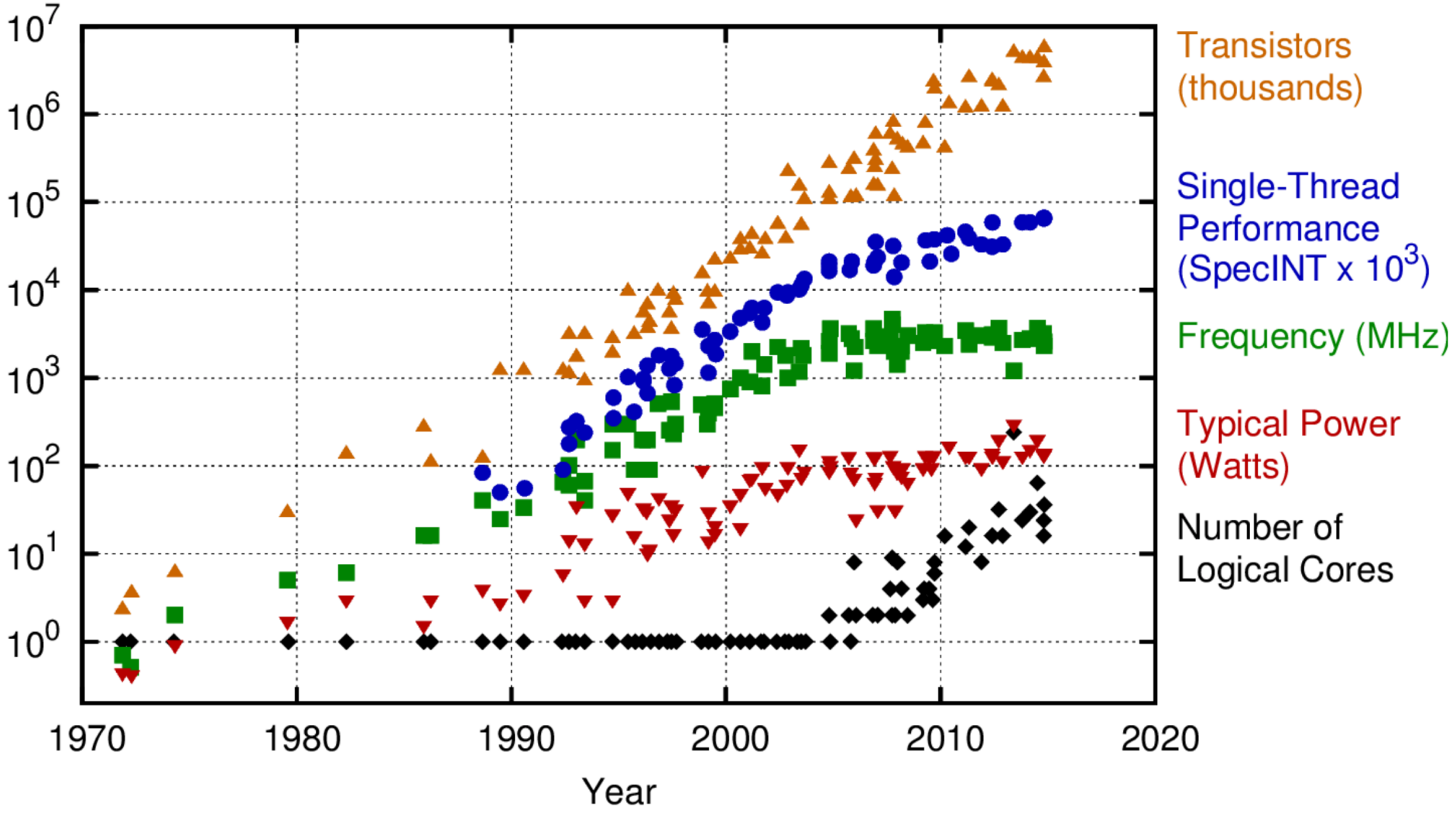
Power and Heat Stall Clock Frequencies

May 17, 2004 … Intel, the world's largest chip maker, publicly acknowledged that it had hit a "thermal wall" on its microprocessor line. As a result, the company is changing its product strategy and disbanding one of its most advanced design groups. Intel also said that it would abandon two advanced chip development projects …

Now, Intel is embarked on a course already adopted by some of its major rivals: obtaining more computing power by stamping multiple processors on a single chip rather than straining to increase the speed of a single processor … Intel's decision to change course and embrace a "dual core" processor structure shows the challenge of overcoming the effects of heat generated by the constant on-off movement of tiny switches in modern computers … some analysts and former Intel designers said that *Intel was coming to terms with escalating heat problems so severe they threatened to cause its chips to fracture at extreme temperatures*…

New York Times

Technology Trends

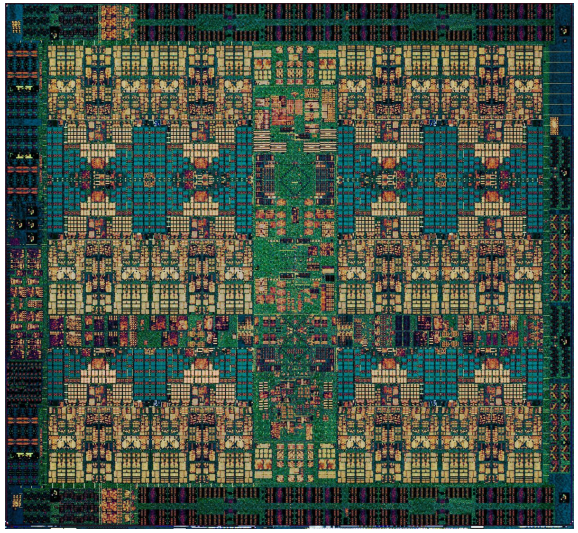


Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

Figure credit: Karl Rupp. https://www.karlrupp.net/wp-content/uploads/2015/06/40-years-processor-trend.png.

Recent Multicore Processors

- 2023: AMD Bergambo
  - — 128 cores, 2-way SMT, 256MB L3 cache
- 2019: AMD EPYC 7742
  - —64 cores; 2-way SMT; 256MB cache
- 2017: IBM Power9
  - —24 cores; 4-way SMT; 120MB cache
- 2016: Intel Knight's Landing
  - —72 cores; 4-way SMT; 36MB cache
- 2015: Oracle SPARC M7
  - —32 cores; 8-way SMT; 64MB cache



IBM Power9
https://en.wikichip.org/wiki/ibm/microarchitectures/power9

Application Pull

- **Complex problems require computation on large-scale data**
- **Sufficient performance is available only through massive parallelism**

Computing and Science

"Computational modeling and simulation are among the most significant developments in the practice of scientific inquiry in the 20th century. Within the last two decades, scientific computing has become an important contributor to all scientific disciplines. It is particularly important for the solution of research problems that are insoluble by traditional scientific theoretical and experimental approaches, hazardous to study in the laboratory, or time consuming or expensive to solve by traditional means"

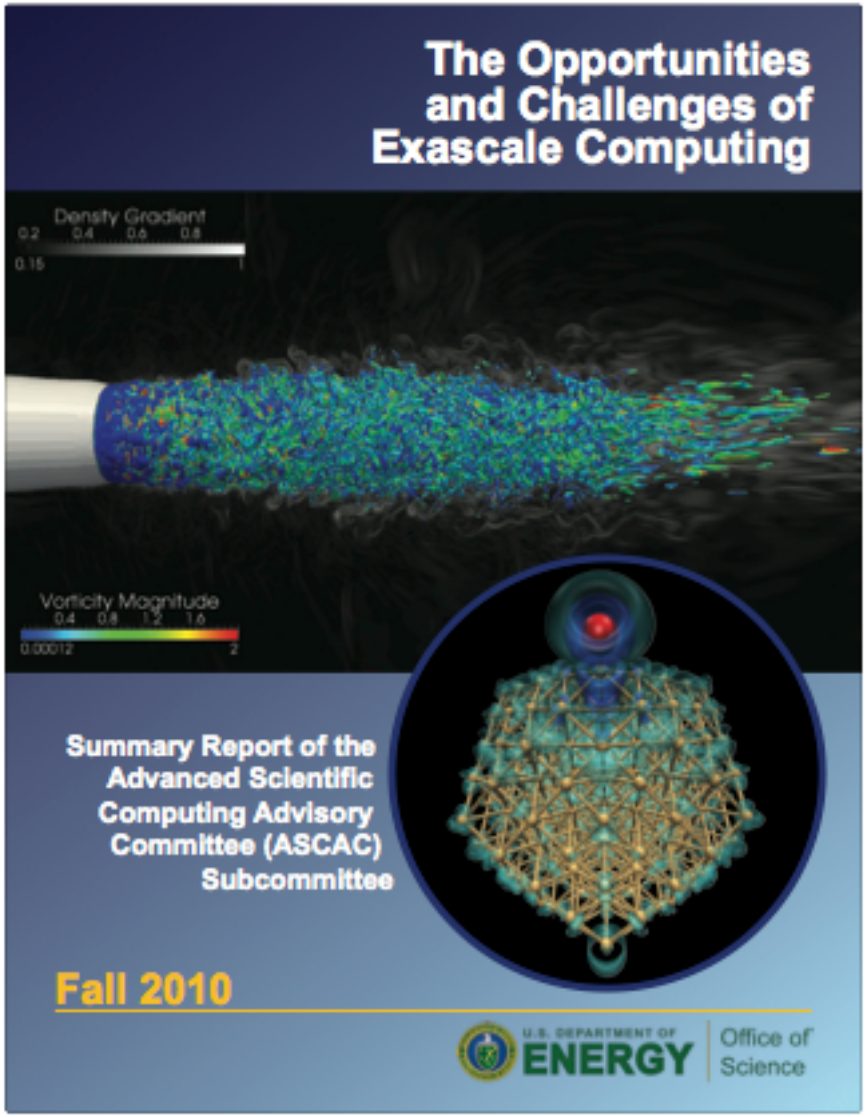
— "Scientific Discovery through Advanced Computing"
DOE Office of Science, 2000

The Need for Speed: Complex Problems

- **Science**
  - —understanding matter from elementary particles to cosmology
  - —storm forecasting and climate prediction
  - —understanding biochemical processes of living organisms
- **Engineering**
  - —multiscale simulations of metal additive manufacturing processes
  - —understanding quantum properties of materials
  - —earthquake and structural modeling
  - —pollution modeling and remediation planning
  - —molecular nanotechnology
- **Business**
  - —computational finance - high frequency trading
  - —information retrieval
  - —data mining “big data”
- **Defense**
  - —nuclear weapons stewardship
  - —cryptology

The Scientific Case for Exascale Computing

- Predict regional climate changes: sea level rise, drought and flooding, and severe weather patterns
- Reduce carbon footprint of transportation
- Improve efficiency and safety of nuclear energy
- Improve design for cost-effective renewable energy resources such as batteries, catalysts, and biofuels
- Certify the U.S. nuclear stockpile
- Design advanced experimental facilities, such as accelerators, and magnetic and inertial confinement fusion
- Understand properties of fission and fusion reactions
- Reverse engineer the human brain
- Design advanced materials



Earthquake Simulation in Japan

Powerful earthquakes leave at least 57 dead, destroy buildings along Japan's western coast

Earthquake Simulation in Japan

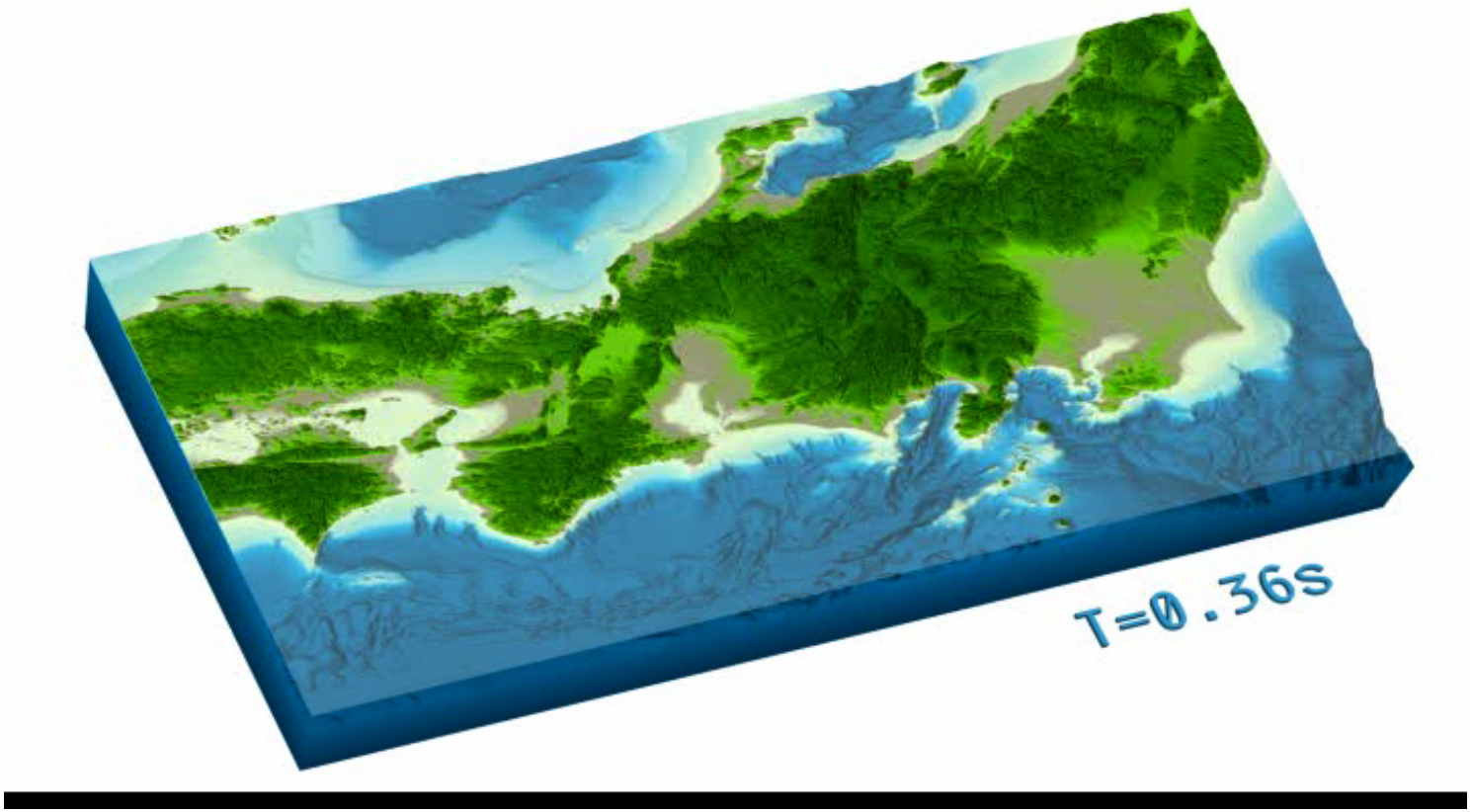
March 11, 2011 Fukushima Daiichi Nuclear Power Plant suffered major damage from a 9.0 earthquake and subsequent tsunami that hit Japan.

The earthquake and tsunami disabled the reactor cooling systems, leading to radiation leaks and triggering a 30 km evacuation zone around the plant.

Confirmed deaths: > 18,500 as of 2021

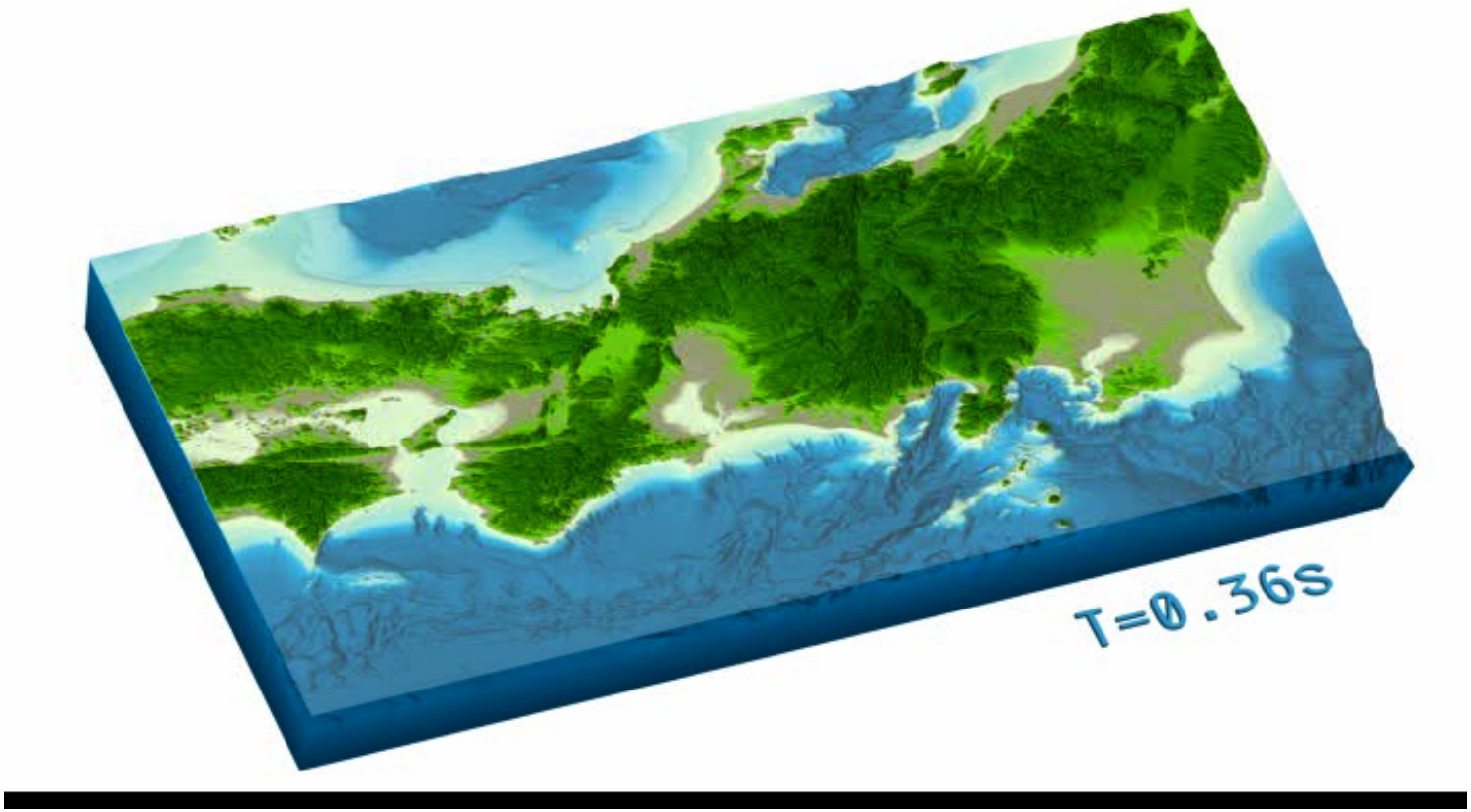
M7.5 Anamizu, Japan - EARTHQUAKE SIMULATION
https://www.youtube.com/watch?v=NSGURtZSu-w

Earthquake Simulation in Japan



M7.5 Anamizu, Japan - EARTHQUAKE SIMULATION
https://www.youtube.com/watch?v=NSGURtZSu-w

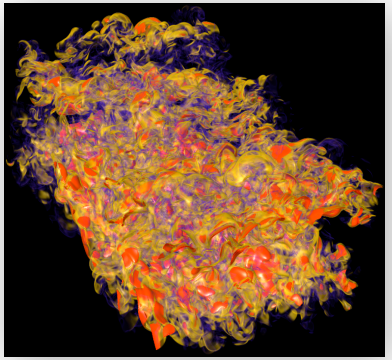
Earthquake Simulation in Japan



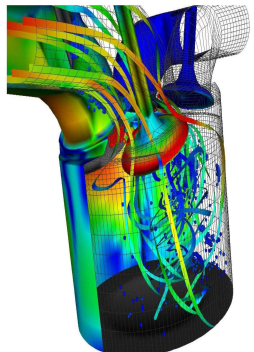
M7.5 Anamizu, Japan - EARTHQUAKE SIMULATION
https://www.youtube.com/watch?v=NSGURtZSu-w

Simulating Turbulent Reacting Flows: S3D

- **Direct numerical simulation (DNS) of turbulent combustion**
  - **state-of-the-art code developed at CRF/Sandia**
    - **PI: Jaqueline H. Chen, SNL**
  - **2020: 600K hours, IBM AC922 2xP9+6xV100**
  - **"DNS of Turbulent Combustion Towards Efficient Engines with In Situ Analytics"**



- **Science**
  - **study micro-physics of turbulent reacting flows**
    - **physical insight into chemistry turbulence interactions**
  - **simulate chemistry and multi-physics (sprays, radiation, soot)**
  - **develop and validate reduced model descriptions used in macro-scale simulations of engineering-level systems**



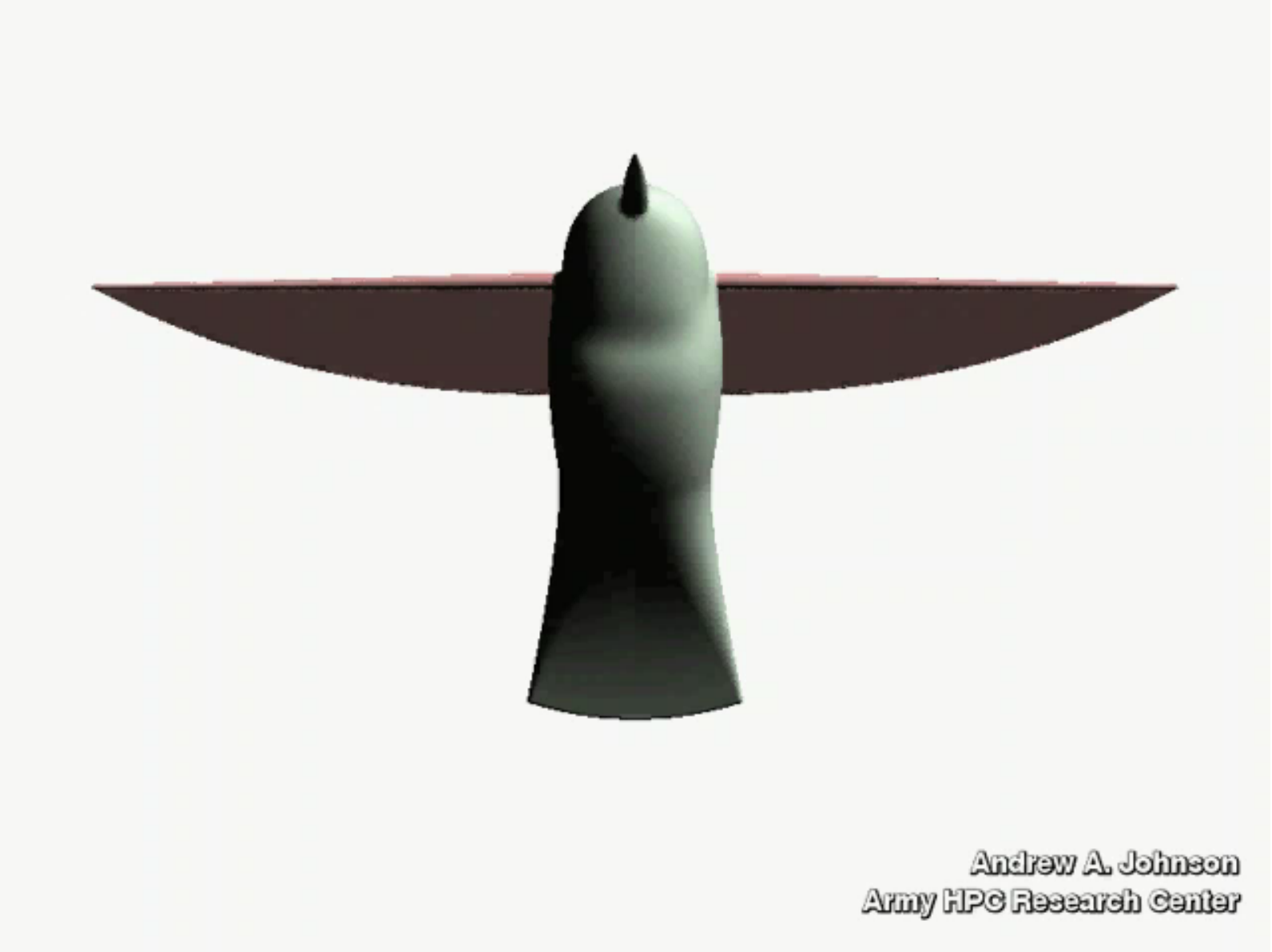
Text and figures courtesy of Jacqueline H. Chen, SNL

Fluid-Structure Interactions

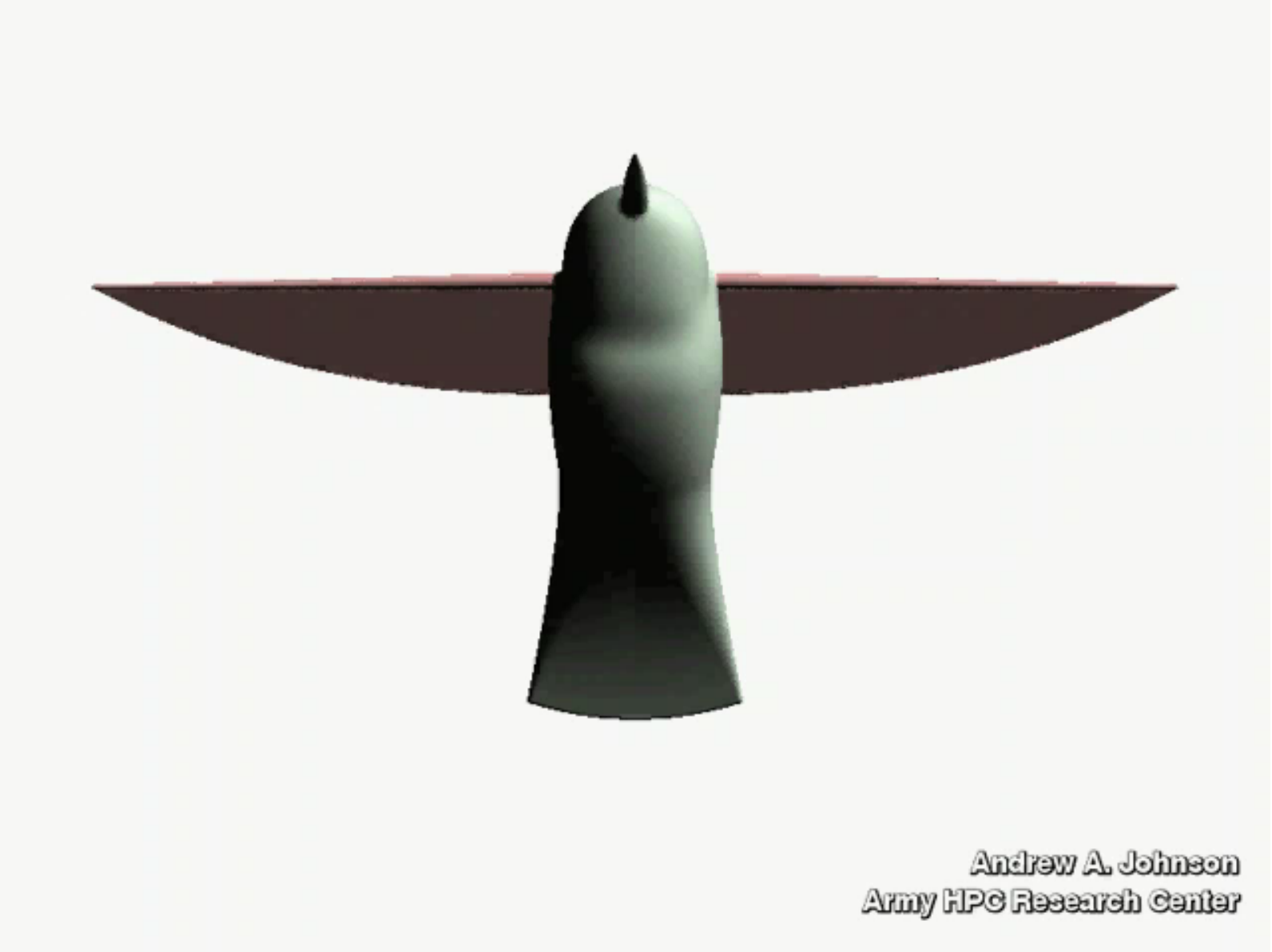
- **Simulate …**
  - —rotational geometries (e.g. engines, pumps), flapping wings
- **Traditionally, such simulations have used a fixed mesh**
  - —drawback: solution quality is only as good as initial mesh
- **Dynamic mesh computational fluid dynamics**
  - —integrate automatic mesh generation within parallel flow solver
    - nodes added in response to user-specified refinement criteria
    - nodes deleted when no longer needed
    - element connectivity changes to maintain minimum energy mesh
  - —mesh changes continuously as geometry + solution changes
- **Example: 3D simulation of a hummingbird's flight**

[Andrew Johnson, AHPCRC 2005]

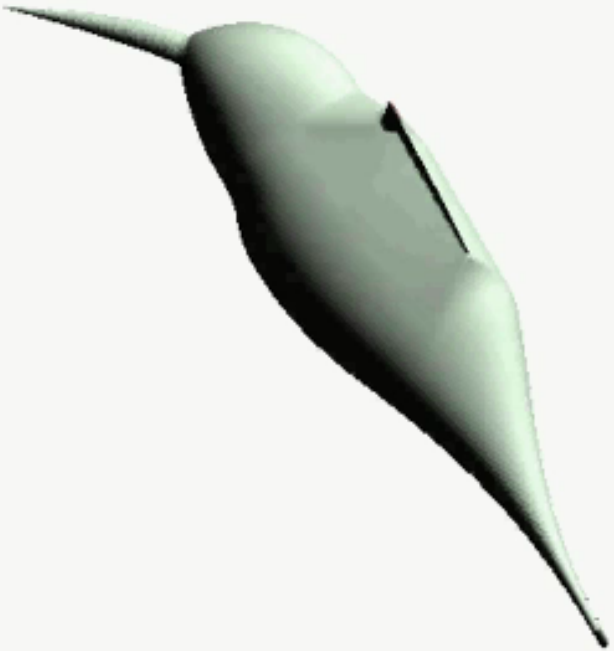
Air Velocity (Front)



Air Velocity (Front)

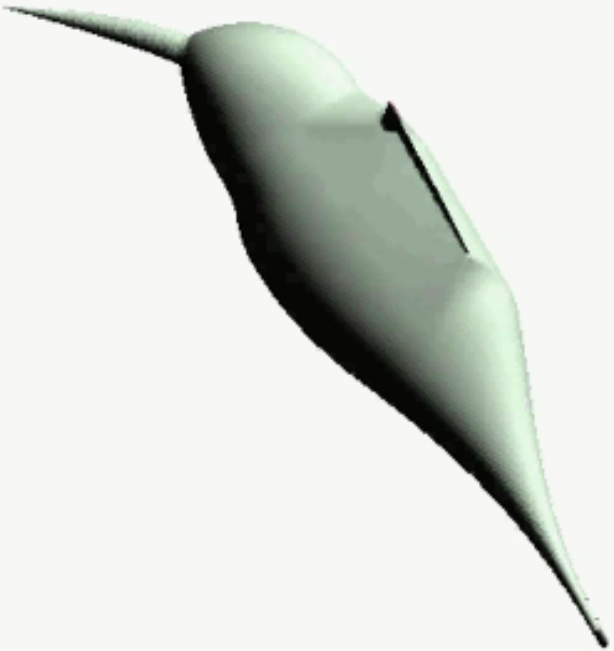


Air Velocity (Side)



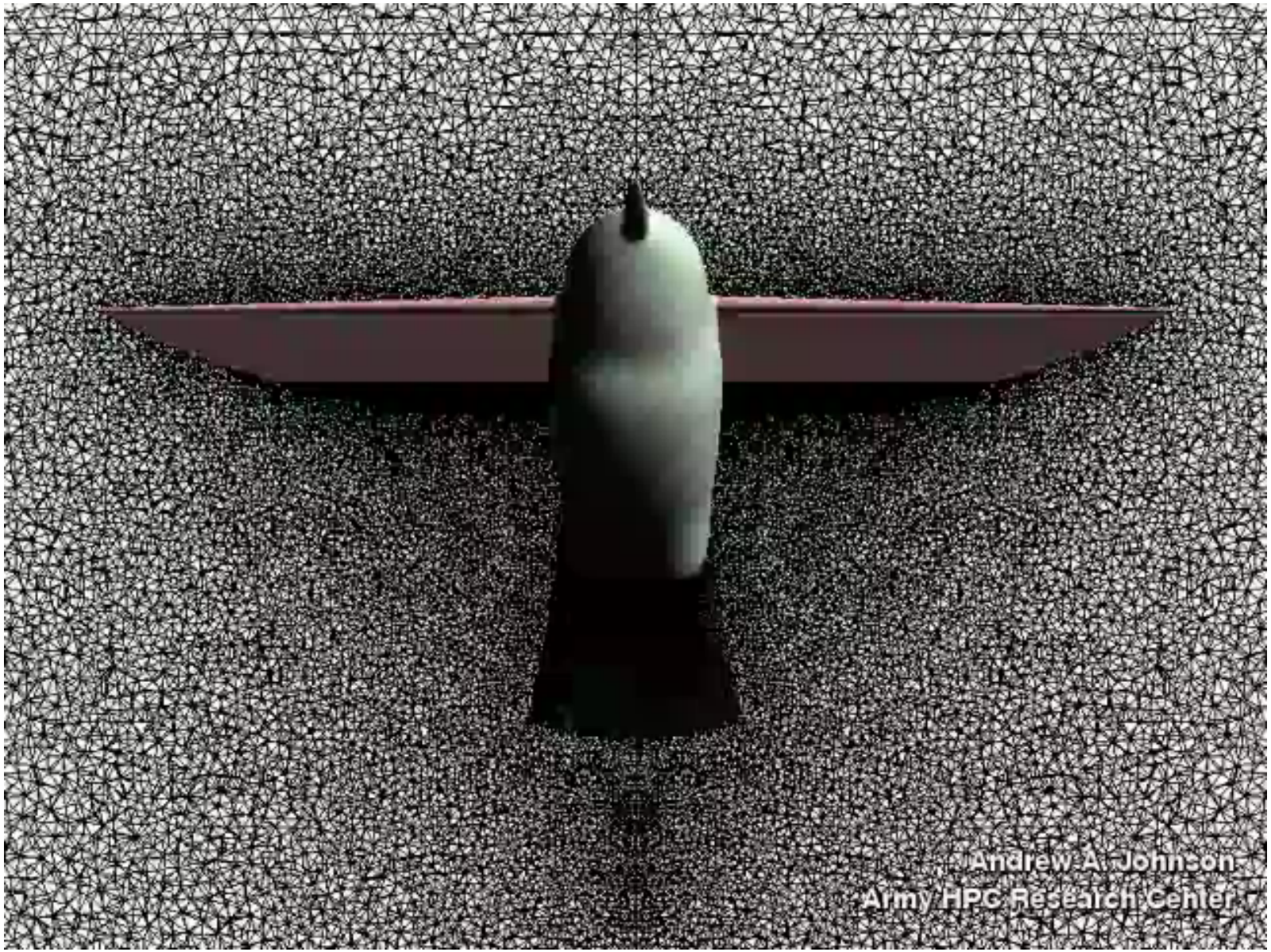
Andrew A. Johnson
Army HPC Research Center

Air Velocity (Side)

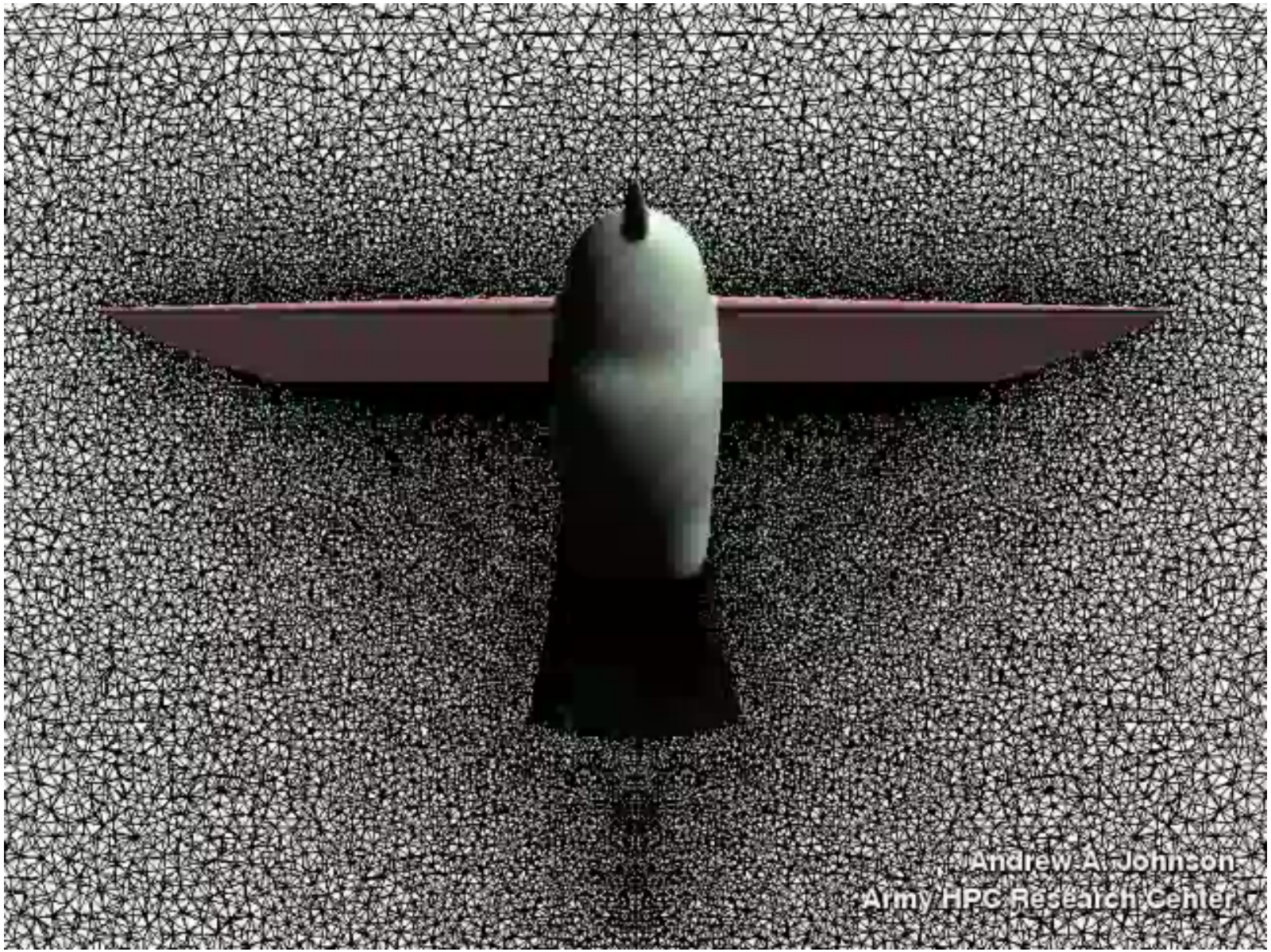


Andrew A. Johnson
Army HPC Research Center

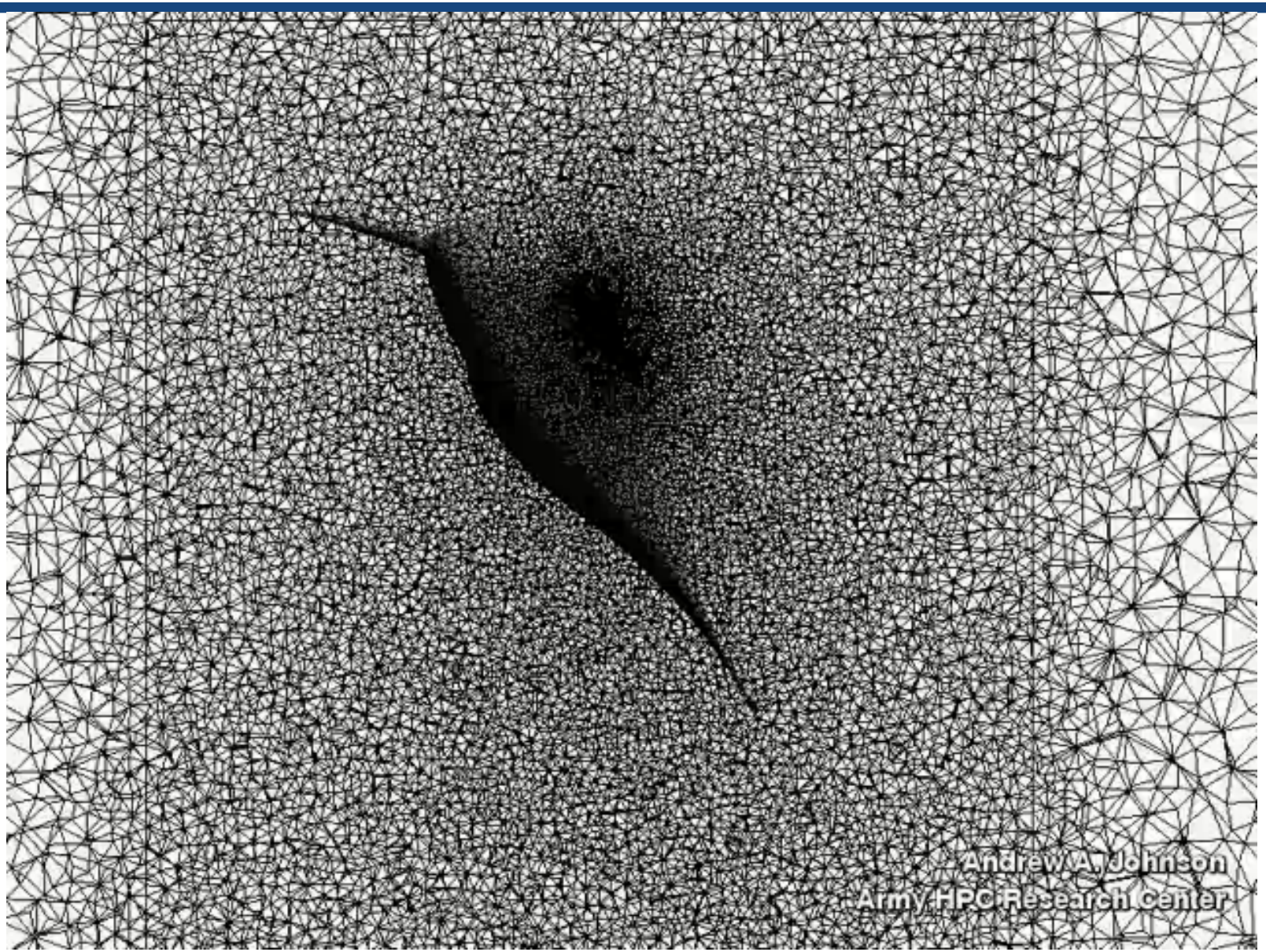
Mesh Adaptation (front)



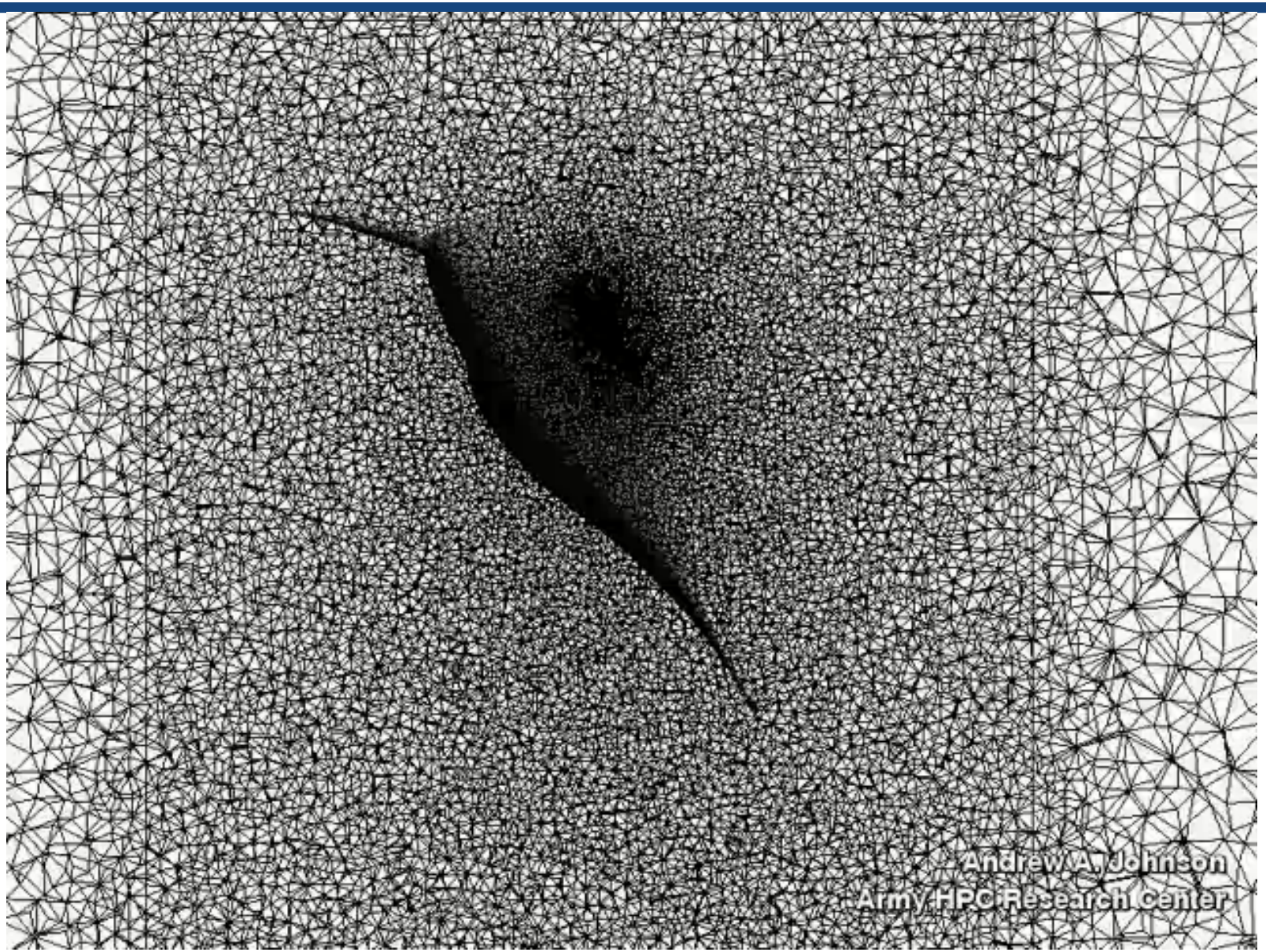
Mesh Adaptation (front)



Mesh Adaptation (side)



Mesh Adaptation (side)



3D Simulations

- **hummingbird's flight**
  - —https://www.youtube.com/watch?v=Dg8xg4U7Xqs
- **3D heart simulation**
  - —https://youtu.be/2LPboySOSvo
- **Airplane wing stall**
  - —https://www.youtube.com/watch?v=gjNPEDBeiTI

Challenges of Explicit Parallelism

- **Algorithm development is harder**
  - —complexity of specifying and coordinating concurrent activities
- **Software development is much harder**
  - —lack of standardized & effective development tools and programming models
  - —subtle program errors: race conditions
- **Rapid pace of change in computer system architecture**
  - —a great parallel algorithm for one machine may not be suitable for another
    - example: homogeneous multicore processors vs. GPUs

Hummingbird Simulation in UPC

- **UPC: PGAS language for scalable parallel systems**
  - —supports a shared memory programming model on a cluster
- **Application overview**
  - —distribute mesh among the processors
  - —partition the mesh among the processors
  - —each processor maintains and controls its piece of the mesh
    - has a list of nodes, faces, and elements
  - —communication and synchronization
    - read-from or write-to other processor's data elements as required
    - processors frequently synchronize using barriers
    - use "broadcast" and "reduction" patterns
  - —constraint
    - only 1 processor may change the mesh at a time

Algorithm Sketch

At each time step…

- **Test if re-partitioning is required**
- **Set up interprocessor communication if mesh changed**
- **Split elements into independent (vectorizable) groups**
- **Calculate the refinement value at each mesh node**
- **Move the mesh**
- **Solve the coupled fluid-flow equation system**
- **Update the mesh to ensure mesh quality**
  - **—swap element faces to obtain a “Delaunay” mesh**
  - **—add nodes to locations where there are not enough**
  - **—delete nodes from locations where there are too many**
  - **—swap element faces to obtain a “Delaunay” mesh**

Parallel Hardware in the Large

ORNL Frontier Supercomputer

Compute Blade (2 nodes)

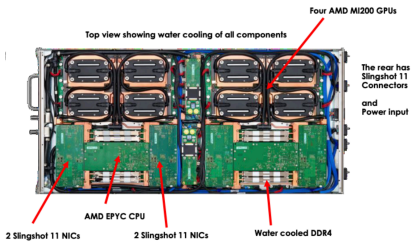
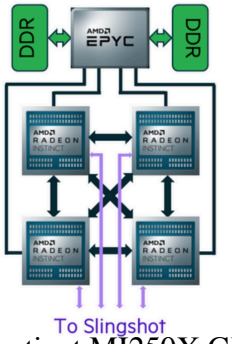


Figure14: One Rack Containing 64 Blades (128 nodes)

System Characteristics	
Node	1 - AMD EPYC 7A53 CPU & 4 - AMD Instinct MI250X GPUs
Nodes per Blade	2
Nodes per Cabinet	128
Nodes per System	9408
Total system memory:	9.2 PB (4.6 PB HBMe2 + 4.6 PB DDR4)
Total on-node NVM	37 PB (66 TB/s read, 62 TB/s write)
Frontier Storage	706 PB (695 PB disk + 11 PB SSD (9.4 TB/s)
Memory Bandwidth between HBM2e and each GPU	3,200 GB/s (3.2 TB/s)
Memory Bandwidth between DDR4 and the CPU	205 GB/s

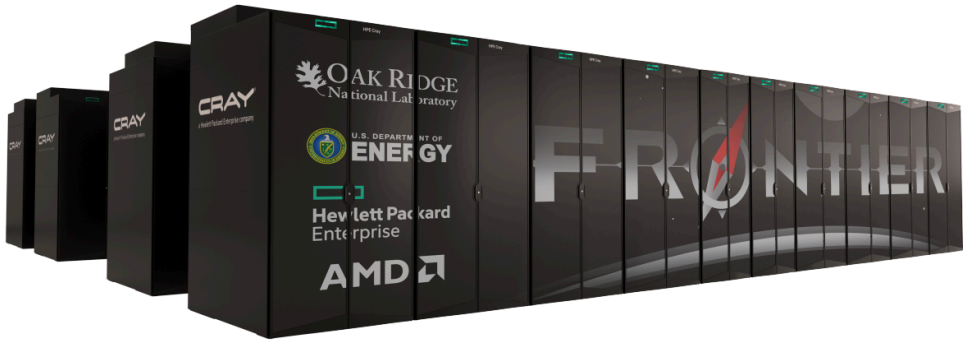
Compute Node

AMD EPYC 64 core @ 2GHz
128 threads



4 x Instinct MI250X GPUs
14,080 stream processor each

1.6 x 10^{18} operations/second = 1.6 ExaFlops



Compute System

Scale of the Largest HPC Systems (Nov 2023)

	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory **United States**	8,699,904	1,194.00	1,679.82
2	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory **United States**	4,742,808	585.34	1,059.33
3	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Microsoft Azure **United States**	1,123,200	561.2	846.84
4	Supercomputer **Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science **Japan**	7,630,848	442.01	537.21
5	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC **Finland**	2,752,704	379.7	531.51
6	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, EVIDEN EuroHPC/CINECA **Italy**	1,824,768	238.7	304.47
7	**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	148.6	200.79
8	**MareNostrum** 5 ACC - BullSequana XH3000, Xeon Platinum 8460Y+ 40C 2.3GHz, NVIDIA H100 64GB, Infiniband NDR200, EVIDEN EuroHPC/BSC **Spain**	680,960	138.2	265.57
9	**Eos** NVIDIA DGX SuperPOD - NVIDIA DGX H100, Xeon Platinum 8480C 56C 3.8GHz, NVIDIA H100, Infiniband NDR400, Nvidia NVIDIA Corporation **United States**	485,888	121.4	188.65
10	**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	94.64	125.71

>400K cores

5 countries

9 are heterogeneous

Scale of the Largest HPC Systems (Nov 2022)

	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)
1	**Frontier** - AMD EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65
2	**Supercomputer Fugaku** - A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21
3	**LUMI** - AMD EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70
4	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos Italy	1,463,616	174.70	255.75
5	**Summit** - IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory USA	2,414,592	148.60	200.79
6	**Sierra** - IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband,DOE/NNSA/LLNL United States	1,572,480	94.64	125.71
7	**Sunway TaihuLight** -, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93.01	125.44
8	**Perlmutter** - AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, HPE DOE/SC/LBNL/NERSC United States	761,856	70.87	93.75
9	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63.46	79.22
10	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, NUDT China	4,981,760	61.44	100.68

Source https://www.top500.org

Scale of the Largest HPC Systems (Nov 2022)

	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)
1	**Frontier** - AMD EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65
2	**Supercomputer Fugaku** - A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21
3	**LUMI** - AMD EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70
4	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos Italy	1,463,616	174.70	255.75
5	**Summit** - IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory USA	2,414,592	148.60	200.79
6	**Sierra** - IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband,DOE/NNSA/LLNL United States	1,572,480	94.64	125.71
7	**Sunway TaihuLight** -, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93.01	125.44
8	**Perlmutter** - AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, HPE DOE/SC/LBNL/NERSC United States	761,856	70.87	93.75
9	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63.46	79.22
10	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, NUDT China	4,981,760	61.44	100.68

all 10 > 550K cores

Source https://www.top500.org

Scale of the Largest HPC Systems (Nov 2022)

	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)
1	**Frontier** - AMD EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65
2	**Supercomputer Fugaku** - A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21
3	**LUMI** - AMD EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70
4	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos Italy	1,463,616	174.70	255.75
5	**Summit** - IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory USA	2,414,592	148.60	200.79
6	**Sierra** - IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband,DOE/NNSA/LLNL United States	1,572,480	94.64	125.71
7	**Sunway TaihuLight** -, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93.01	125.44
8	**Perlmutter** - AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, HPE DOE/SC/LBNL/NERSC United States	761,856	70.87	93.75
9	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63.46	79.22
10	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, NUDT China	4,981,760	61.44	100.68

all 10 > 550K cores

> 2M cores

Source https://www.top500.org

Scale of the Largest HPC Systems (Nov 2022)

	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)
1	**Frontier** - AMD EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65
2	**Supercomputer Fugaku** - A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21
3	**LUMI** - AMD EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70
4	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos Italy	1,463,616	174.70	255.75
5	**Summit** - IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory USA	2,414,592	148.60	200.79
6	**Sierra** - IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband,DOE/NNSA/LLNL United States	1,572,480	94.64	125.71
7	**Sunway TaihuLight** -, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93.01	125.44
8	**Perlmutter** - AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, HPE DOE/SC/LBNL/NERSC United States	761,856	70.87	93.75
9	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63.46	79.22
10	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, NUDT China	4,981,760	61.44	100.68

all 10 > 550K cores

> 2M cores

heterogeneous manycore

Source https://www.top500.org

Achieving High Performance on Parallel Systems

Computation is only part of the picture

- **Memory latency and bandwidth**
  - —CPU rates are > 200x faster than memory
  - —bridge speed gap using memory hierarchy
  - —more cores exacerbates demand
- **Interprocessor communication**
- **Input/output**
  - — I/O bandwidth to disk typically needs to grow linearly with the # processors
  - — increasing using SSD for acceleration

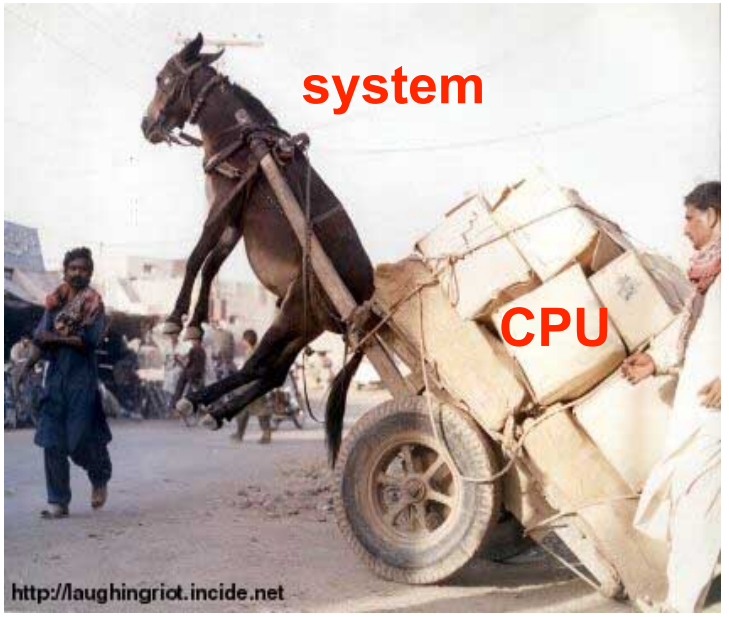


Image Credit: Bob Colwell, ISCA 1995

Challenges of Parallelism in the Large

- **Parallel science applications are often very sophisticated**
  - —e.g. adaptive algorithms may require dynamic load balancing
- **Multilevel parallelism is difficult to manage**
- **Extreme scale exacerbates inefficiencies**
  - —algorithmic scalability losses
  - —serialization and load imbalance
  - —communication or I/O bottlenecks
  - —insufficient or inefficient parallelization
- **Hard to achieve top performance even on individual nodes**
  - —contention for shared memory bandwidth
  - —memory hierarchy utilization on multicore processors

Next Class

- Introduction to parallel algorithms
  - —tasks and decomposition
  - —task dependences and critical path
  - —mapping tasks
- Decomposition techniques
  - —recursive decomposition
  - —data decomposition

Parallel System for the Course

- **GPU Cluster @ IIT Dharwad**
  - **Specs**
- **HPC Cluster @ IIT Dharwad**
  - **Specs**