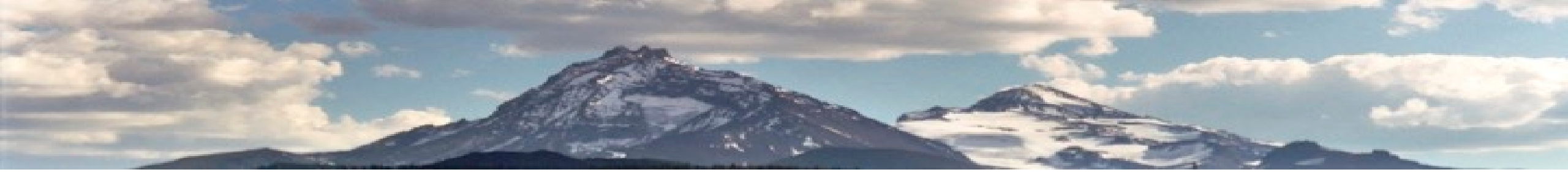


Perspectives on Supercomputing: Past, Present, and Future

Prof. Allen D. Malony
Department of Computer Science
University of Oregon

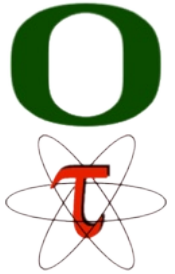
Digital Futures Seminar
October 2, 2025

KTH VETENSKAP OCH KONST

digital futures

Who am I?

- Member of Technical Research staff, HP Labs (1981-1986)
- Ph.D., University of Illinois, Urbana-Champaign (1991)
  - Center for Supercomputing R&D (Cedar Project) (1987–1991)
  - Ph.D. thesis on “Performance Observability”
- Professor, University of Oregon (UO) (since 1992)
  - Department of Computer Science
  - Lead, *TAU Project* (1992-present)
  - Director, Oregon Advanced Computing Institute for Science and Society
  - CEO and Director, ParaTools, Inc. (since 2004)
- Recognition
  - Fulbright Research Scholar: The Netherlands (1991), Austria (1998)
  - Alexander von Humboldt Research Award for Senior U.S. Scientists (2002)
  - Fulbright-Tocqueville Distinguished Chair, France (2016)
  - Fulbright-Nokia Distinguished Chair in ICT, Finland (2021)



ParaTools

Where do I come from?

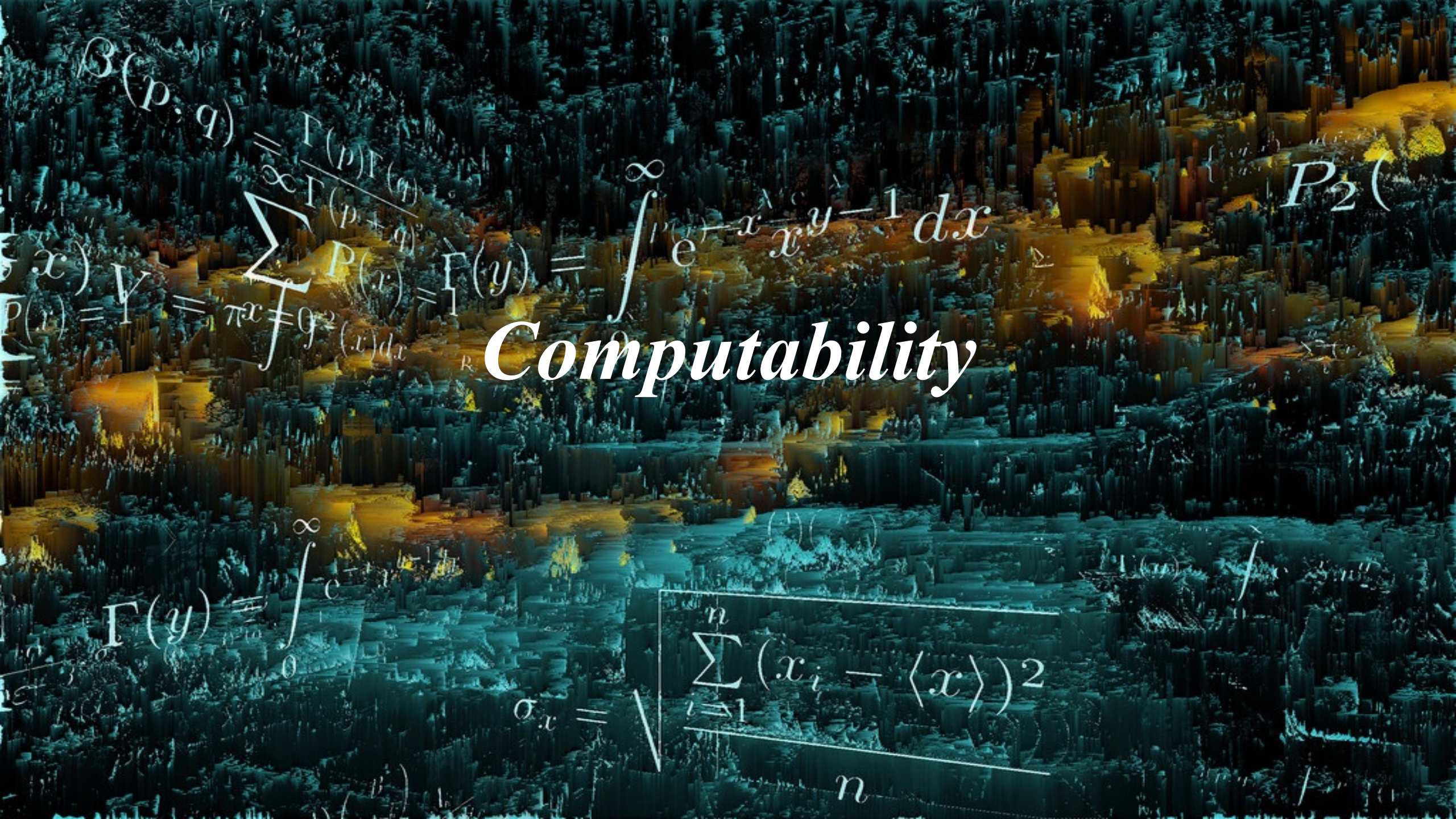
Oregon, USA

Seminar Outline

- Computability and Complexity
- Parallelism and Performance
- Supercomputing
- Parallel performance research
- Future directions
- Takeaway Thoughts

- Special surprise

Perspectives are in the context of what has motivated me in my scholarly research activities during my career


Computability

Decision Problem (Entscheidungsproblem)

- In 1900, David Hilbert published 23 unsolved problems
  - Was mathematics complete?
  - Was mathematics consistent?
  - Was mathematics decidable? (the *Entscheidungsproblem*)
  - Hilbert's 10th problem
    - determine the solvability of a Diophantine equation
    - became a question about a "decision method"
- Most general form of the *Entscheidungsproblem:*

  "A quite definite generally applicable prescription is required which will allow one to decide in a finite number of steps the truth or falsity of a given purely logical assertion …"

An Effective Method

- What is a *definite generally applicable prescription*?
  - Process for solving a (particular class of) problem
  - What is generally called in mathematics an *effective method*
- Alonzo Church created *λ-calculus* (1936)
  - Defined the concept of *effective calculability*
  - Proved that a general solution to the decision problem is impossible
- Alan Turing invented the *Turing Machine* (1936)
  - Showed how to construct an effective method
  - Applied it to the problem of computable numbers

    "… a number is computable if its decimal can be written down by a machine."
  - Proved the decision problem is impossible

A.M. Turing, "On Computable Numbers, with an Application to the Entscheidungs Problem," 1936.

Nature of Computability (Church-Turing Thesis)

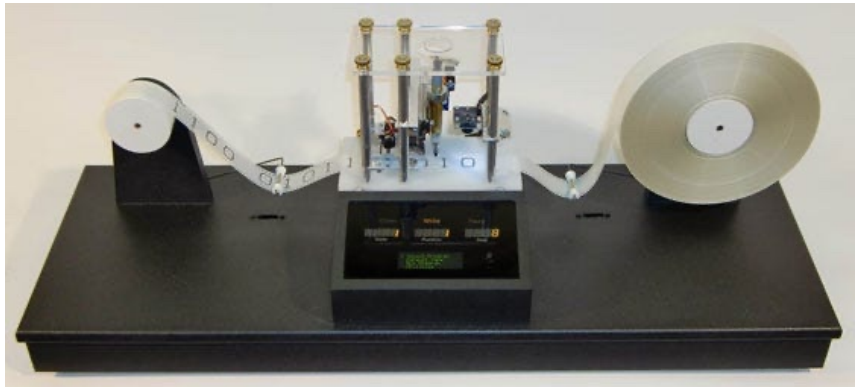
- A Turing Machine (TM) is a *model of computation*
  - It is a universal model (*Universal Turing Machine*)
    - encode a TM and feed it to a Universal Turing Machine
    - a Turing Machine considered as data (this is a profound concept)
- *Church-Turing Thesis*
  - There is no effective model of computing more powerful
  - What is effectively calculable is computable
  - What is "computable?" It is a Turing Machine!
- What problems are computable?
  - Interestingly, not all problems are computable

Turing Machine and the "Computer"

- Turing imagines not a mechanism, but a person ("computer") who executes deterministic mechanical rules "in a desultory manner":

  "Let us imagine the operations performed by the computer to be split up into 'simple operations' which are so elementary that it is not easy to imagine them further divided. Every such operation consists of some change of the physical system consisting of the computer and his tape."

- Turing described how to construct a machine to be the "computer"
- Deterministic rules are *algorithms*



https://spectrum.ieee.org/032610-diy-turing-machine

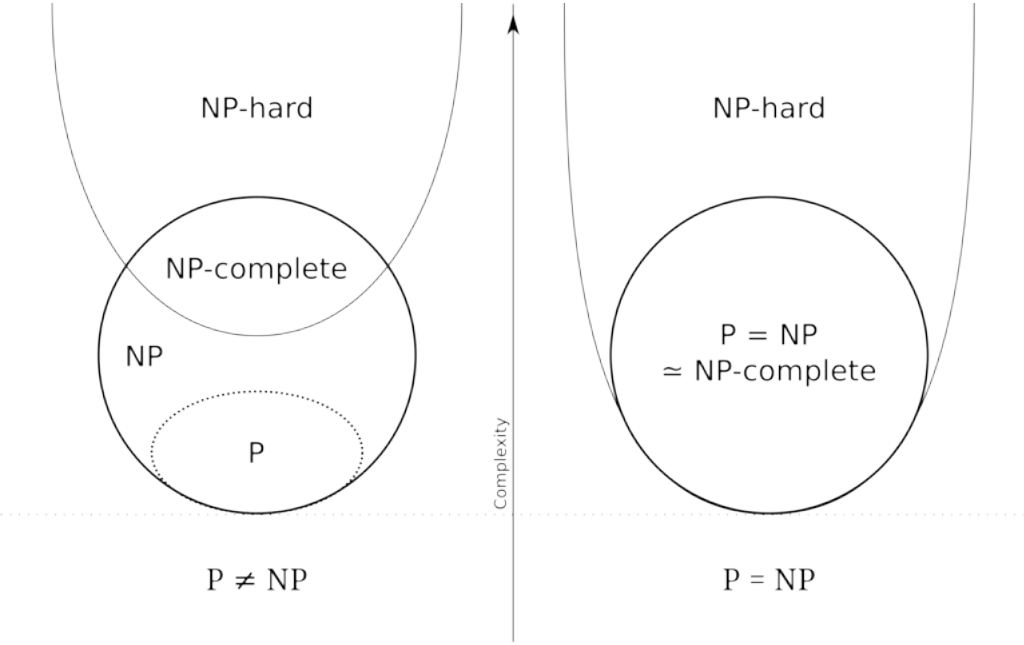

Complexity

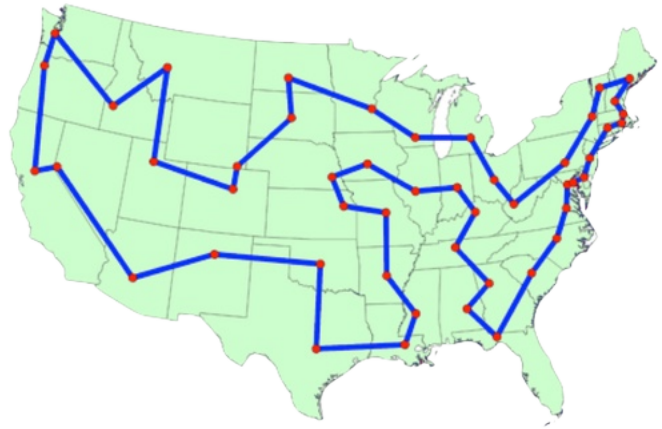
Computability versus Complexity

- *Computational power* has to do with *what* can be computed
- Turing Machines are as powerful as any computer in existence
  - Simply put, it can solve the same computational problems
- Computability is not the same as complexity
  - A problem just needs an algorithm to be computable
- *Complexity* has to do with how *hard* it is to solve a problem
  - Measure of the difficulty of the problem solution (running the algorithm)
- A *complexity theory* gives a way to talk about this
  - Metrics of "time" (T = # operations) and "space"
  - Complexity classes based on how problems scale

Complexity Classes

- Some computational problems are harder to solve
- *P:*
  - Computational problems solvable in *polynomial time* (# operations) by a TM
- *NP:*
  - Computational problems solvable in *polynomial time* by a *nondeterministic* TM
- *NP-complete:*
  - A problem *p* in NP is NP-complete if every other problem in NP can be reduced into *p* in polynomial time
  - Problem *p* is as hard as all other NP problems
- *NP-hard* problems are as hard as all NP problems
  - The traveling salesman problem is NP-hard
- P $\stackrel{?}{=}$ NP is a major unsolved problem in CS





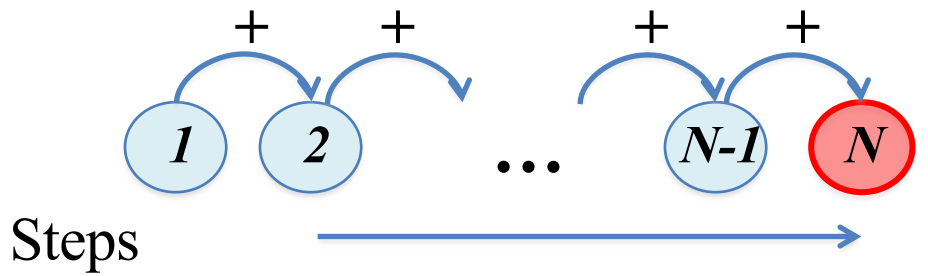

Parallelism

Complexity and Parallelism

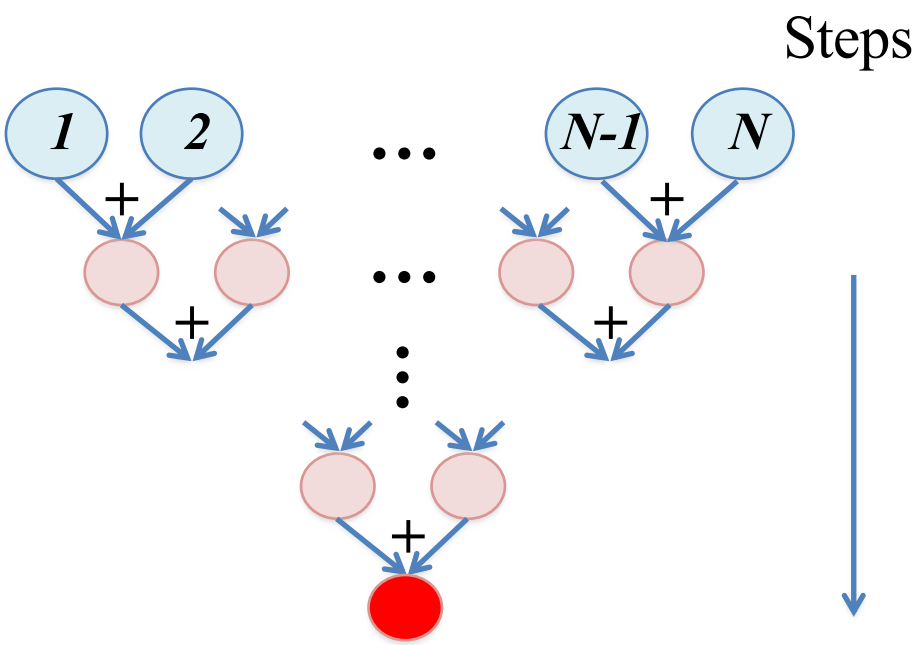
- Complexity is about what happens when problems scale!
- Algorithm complexity has to do with the # operations
  - *Sequential step*: only one operation can be done by a step
  - *Sequential algorithm*: only one operation at any step
- Algorithms can be *parallel*
  - At any step, multiple operations could possibly be done (no dependencies)
  - Effectively reduces total # steps (it does not change complexity class!)
- Algorithms have different parallelism characteristics and limits
  - Parallel behavior (*parallel profile*) can also change during execution
- Dependencies between algorithm operations regulate parallelism

Consider Adding N Numbers

- Sequential algorithm will add the numbers in order
  - Let T_{seq} be # sequential steps
  - N-1 adds
  - $T_{seq} = N-1$

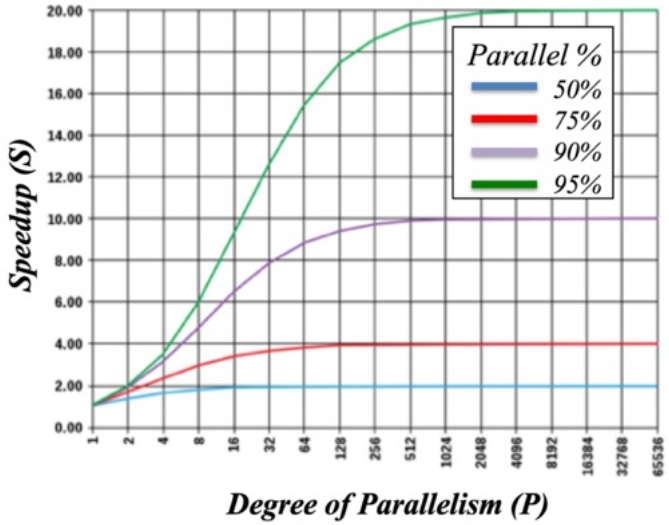


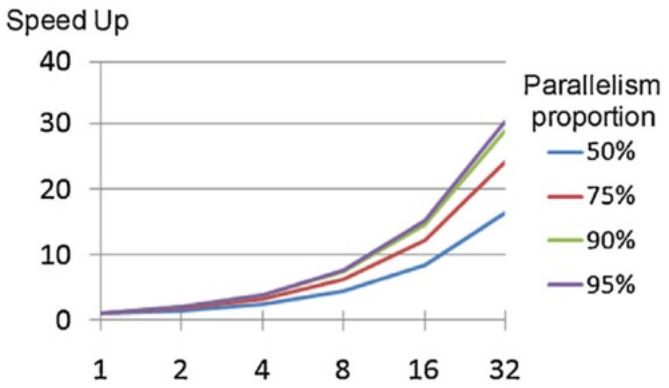
- Can this algorithm be "parallelized"? Yes!
  - At each step, compute a partial sum
  - Construct a binary tree of partial sums
  - $\log_2 N$ levels in tree
  - Let T_{par} be # parallel steps
  - $T_{par} = \log 2\ N$ (= 10 for N=1024)

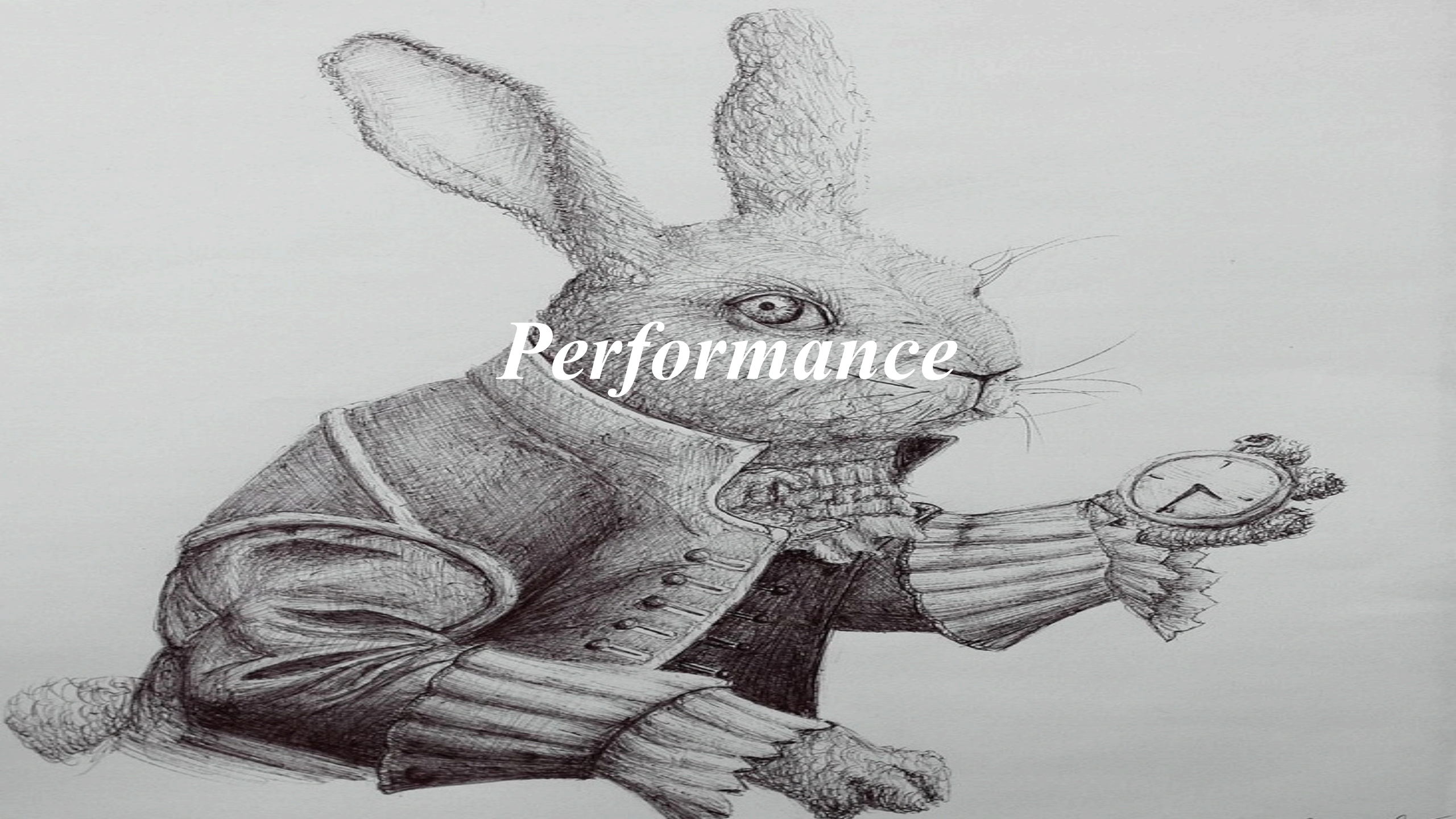


Parallel Speedup Models

- *Speedup* is the algorithmic improvement from parallelism
  - As the effectual degree of parallel operation increases
  - With respect to the *problem size* (i.e., work)
- *Amdahl's Law*
  - Assumes a fixed problem size
  - Speedup is asymptotically bounded (*strong scaling*)
- *Gustafson-Barsis' Law*
  - Problem size is allowed to increase with parallel degree
  - Speedup can continue to improve (*weak scaling*)
- Other models:
  - Work-Span, Embarassingly Parallel, …
- Algorithms determine parallelism complexity and hence speedup potential






Performance

Complexity versus Performance

- Complexity only says how hard the problem is
  - It does not say how an algorithm will perform on a real computer system
- Parallelism only offers speedup opportunities
- What is performance?
  - Computational requirements (What needs to be done?)
  - Computing resources (What it costs to do it?)
- Performance itself is a measure of how well the computational requirements can be satisfied by available resources
- We evaluate performance to understand the relationships between requirements and resources

Performance Metrics and Rates

- How do we quantify computational requirements?
- A computer does *work* by executing instructions
  - Logical operations
  - Arithmetic operations
- Performance can be measured by operational rates
  - Instructions per second (*IPS*)
  - Arithmetic operations per seconds (*OPS*)
  - Scientific computations use floating point (*FLOPS*)
- *Peak performance* is what a machine is capable of
- *Achieved performance* is what is actually delivered
  - Depends on both machine hardware and program (algorithm)


Supercomputing

What is a supercomputer?

- Computational power is realized by a real computer
- A *high-performance computing (HPC)* machine uses state-of-the-art technology to achieve high computational potential
- A *supercomputer* is considered to be an HPC system that attempts to achieve the highest performance possible
- What drives HPC and supercomputing?
  - It's all about parallelism and parallel processing!
  - Supercomputers are parallel systems
- How do we quantify their computational potential?
- How do we evaluate performance on supercomputer systems?

Phases of Supercomputing (Parallel) Architecture

- **1950s:** sequential instruction execution
- **1960s:** instruction level parallelism (ILP)
- **1970s:** vector processors, parallel memory
- **1980s:** SIMD, shared-memory multiprocessors (SMP)
- **1990s:** scalable SMPs, distributed memory systems, MPP
- **2000s:** multiple cores, interconnection networks
- **2010s:** many cores, GPUs, heterogeneity, larger scale
- **2020s:** exascale, accelerators, ML/AI, quantum, ...

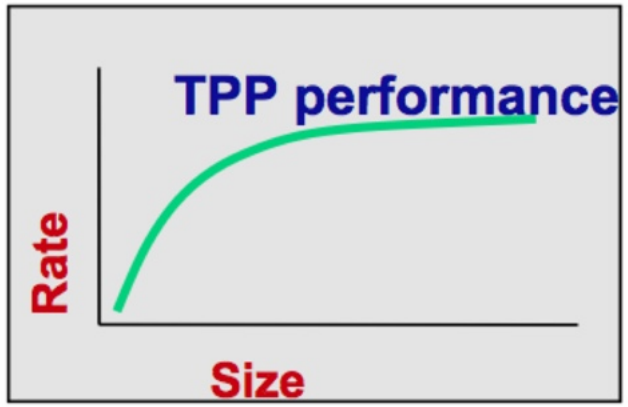
Architectural advances enabled by technology

Parallelism and Scalability

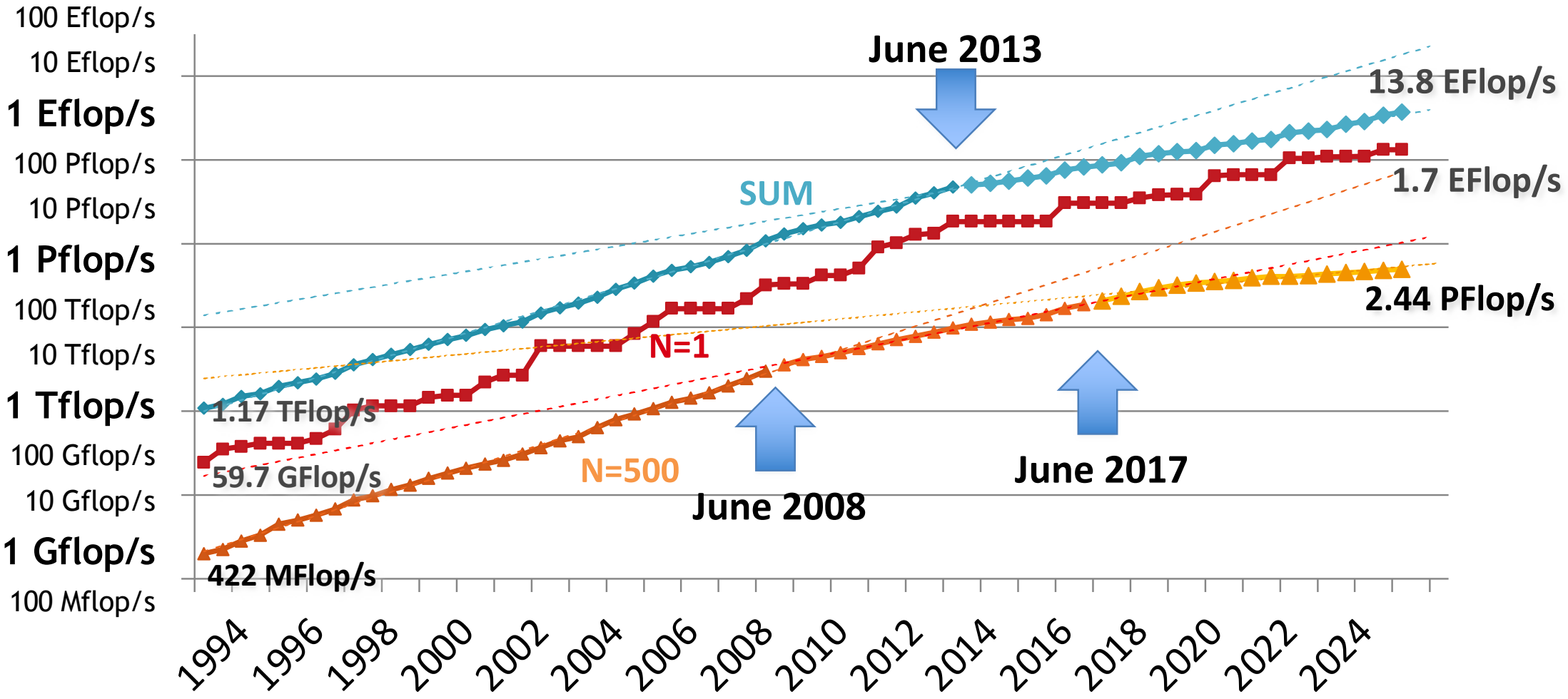
- A parallel program utilizes parallel components of HPC system
- A parallel program can scale in:
  - Amount of computational work to perform
  - Degree of parallelism used
- How do you evaluate scalability?
- Comparative evaluation
  - Peak performance capability of machine
  - Relative performance at different program scales
- Use parallel efficiency measure
- Apply performance metrics and rates

Top500 Performance Benchmarking Methodology

- Listing of the world's 500 supercomputers
- Yardstick for high-performance computing
- Benchmark problem (*Linpack* benchmark) (HPL)
  - Solve a dense linear system of equations ($Ax = b$)
  - Scale problem size to achieve maximum FLOPS/sec
- Report
  - *Rmax* : maximal performance
  - *Rpeak* : theoretical peak performance (TPP)
  - *Nmax* : problem size needed to achieve Rmax
  - N1/2 : problem size needed to achieve 1/2 of Rmax
- Updated twice a year at SC and ISC conferences



Top500 – HPL Performance History (June 2025)

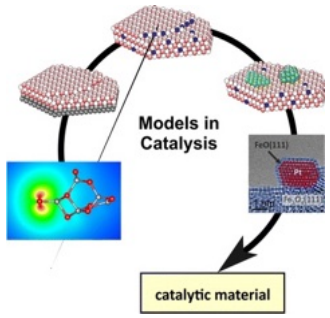


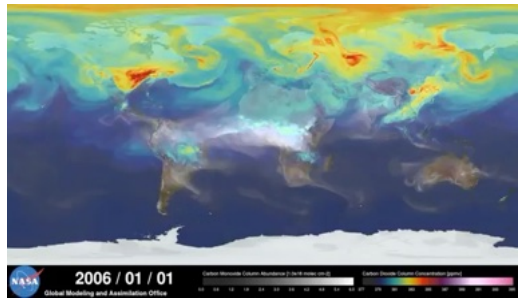
Top 10 – HPL Performance (June 2025)

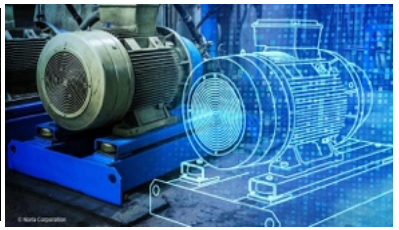
#	Site	Manufacturer	TOP10 Computer of the TOP500	Country	Cores	Rmax [Pflops]	Power [MW]	Peak	Nmax
1	Lawrence Livermore National Laboratory	HPE	El Capitan HPE Cray EX255a, AMD EPYC 24C 1.8GHz, Instinct MI300A, Slingshot-11	USA	11,039,616	1,742	29.6	2,746	25,446,528
2	Oak Ridge National Laboratory	HPE	Frontier HPE Cray EX235a, AMD EPYC 64C 2.0GHz, Instinct MI250X, Slingshot-11	USA	9,066,176	1,353	24.6	2,056	24,837,120
3	Argonne National Laboratory	Intel	Aurora HPE Cray EX/Intel Exascale Compute Blade, Xeon Max 9470, Data Center GPU Max, Slingshot-11	USA	4,742,808	1,012	38.7	1,980	28,773,888
4	EuroHPC / FZJ	EVIDEN	JUPITER Booster BullSequana XH3000, NVIDIA GH200 Superchip, InfiniBand NDR	Germany	4,801,344	793.4	13.1	930	19,668,992
5	Microsoft Azure	Microsoft	Eagle Microsoft NDv5, Xeon Platinum 8480C, NVIDIA H100, Infiniband NDR	USA	1,123,200	561.2		847	11,796,480
6	Eni S.p.A. Center for Computational Science	HPE	HPC6 HPE Cray EX235a, AMD EPYC 64C 2.0GHz, Instinct MI250X, Slingshot-11	Italy	3,143,520	477.9	8.5	607	13,914,880
7	RIKEN Center for Computational Science	Fujitsu	Fugaku Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D	Japan	7,630,848	442.0	29.9	537	21,288,960
8	Swiss National Supercomputing Centre (CSCS)	HPE	Alps HPE Cray EX254n, NVIDIA Grace 72C 3.1GHz, GH200, Slingshot-11	Switzerland	2,121,600	434.9	7.1	575	15,400,960
9	EuroHPC / CSC	HPE	LUMI HPE Cray EX235a, AMD EPYC 64C 2.0GHz, Instinct MI250X, Slingshot-11	Finland	2,752,704	379.7	7.1	532	13,685,760
10	EuroHPC / CINECA	EVIDEN	Leonardo Atos BullSequana XH2000, Xeon 32C 2.6GHz, NVIDIA A100, HDR Infiniband	Italy	1,824,768	241.2	7.5	306	10,229,760

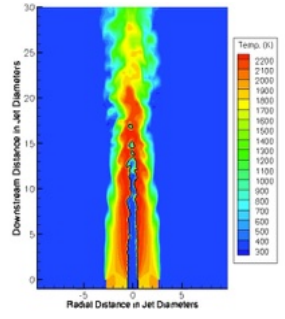
What are these supercomputing systems doing?

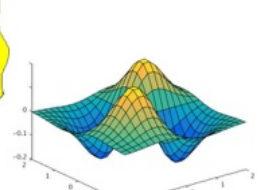
- Modeling and Simulation (ModSim)
- Domains
  - Climate
  - Combustion
  - Fusion
  - Catalysis
  - Supernovae
  - Materials
  - Digital twins
- Core computations:
  - 3D PDEs usually
  - Sparse matrices (not dense)

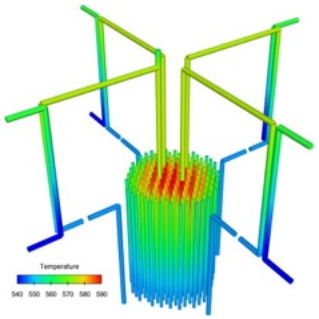


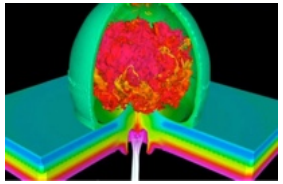


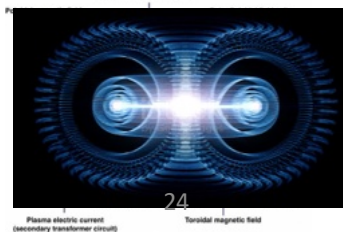






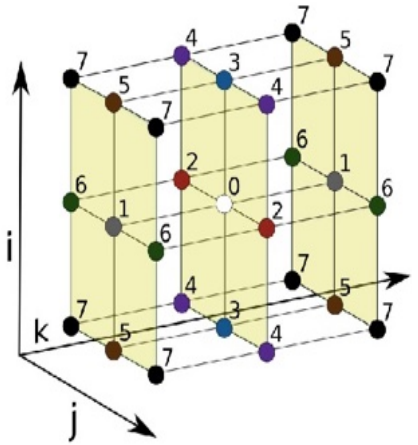






Is HPL still relevant? HPCG – the other benchmark

- Linpack has been a benchmark standard since 1979
  - Easy to understand and captures certain trends
- Technology and HPC architecture have changed
  - Arithmetic was expensive and now inexpensive
  - Data movement (memory, interconnect) more important factor
- Linpack is no longer strongly correlated to real scientific applications
- High Performance Conjugate Gradient (HPCG)
  - Solves $Ax = b$ where A is large and sparse
  - Computational and communication patterns prevalent in variety of methods for discretization and numerical PDEs



Top 10 – HPCG Performance (June 2025)

#	T	Site	Manufacturer	TOP10 Computer for HPCG	Country	HPCG [Pflop/s]	HPL [Pflop/s]	HPCG/ Peak	HPCG/ HPL
1	1	Lawrence Livermore National Laboratory	HPE	El Capitan HPE Cray EX255a, AMD EPYC 24C 1.8GHz, Instinct MI300A, Slingshot-11	USA	17.407	1,742.0	0.63%	1.0%
2	7	RIKEN-CCS	Fujitsu	Fugaku Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D	Japan	16.005	442.0	3.0%	3.6%
3	2	Oak Ridge National Laboratory	HPE	Frontier HPE Cray EX235a, AMD EPYC 64C 2.0GHz, Instinct MI250X, Slingsh.-10	USA	14.054	1,206.0	0.8%	1.2%
4	3	Argonne National Laboratory	Intel	Aurora HPE Cray EX/Intel Exascale Compute Blade, Xeon Max 9470, Data Center GPU Max, Slingshot-11	USA	5.613	1,012.0	0.3%	0.6%
5	9	EuroHPC / CSC	HPE	LUMI HPE Cray EX235a, AMD EPYC 64C 2.0GHz, Instinct MI250X, Slingsh.-10	Finland	4.587	379.7	0.9%	1.2%
6	8	Swiss National Supercomputing Centre (CSCS)	HPE	Alps HPE Cray EX235a, NVIDIA Grace 72C 3.1GHz, GH200, Slingshot-11	Switzer-land	3.671	270.0	1.0%	1.4%
7	10	EuroHPC / CINECA	EVIDEN	Leonardo Atos BullSequana XH2000, Xeon 32C 2.6GHz, NVIDIA A100, HDR Infiniband	Italy	3.114	241.2	1.0%	1.3%
8	15	National Institute of Advanced Industrial Science and Technology (AIST)	HPE	ABCI 3.0 HPE Cray XD670, Xeon 48C 2.1 GHz, NVIDIA H200, Infiniband HDR	Japan	2.446	145.1	1.4%	1.7%
9	25	NERSC - Lawrence Berkeley National Laboratory	HPE	Perlmutter HPE Cray EX235n, AMD EPYC 64C 2.45GHz, NVIDIA A100, Slingshot-10	USA	1.905	79.2	1.7%	2.4%
10	20	Lawrence Livermore National Laboratory	IBM	Sierra IBM Power System, P9 22C 3.1 GHz, Volta GV100, EDR	USA	1.796	94.6	1.4%	1.9%

* It is unclear if HPL Nmax from HPL is used for HPCG

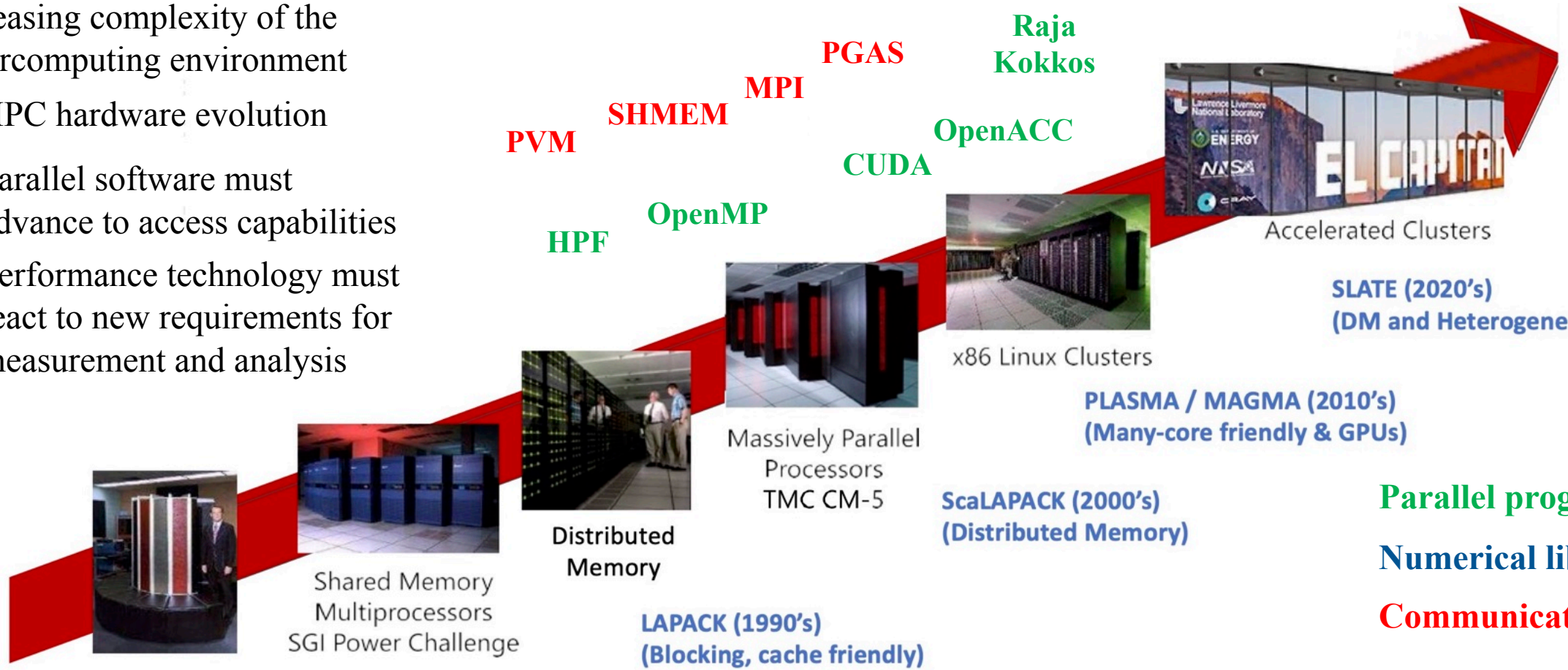
HPC Systems Technology in Top500

- Early in the history of the Top500, most HPC systems were specially built for computational science applications
- Paradigm shift with the "Attack of the Kill Micros"
  - More and more commodity-driven technology
- Top500 now
  - Commodity processors: 93% use x86 (Intel, AMD) instruction set
  - Commodity accelerators: 92% use NVIDIA
  - Commodity interconnect: 85% use Ethernet or Infiniband
  - Commodity OS: 100% run on Linux
- Concern about the "end of Moore's Law"

HPC Software Technology Evolution

Increasing complexity of the supercomputing environment

- HPC hardware evolution
- Parallel software must advance to access capabilities
- Performance technology must react to new requirements for measurement and analysis

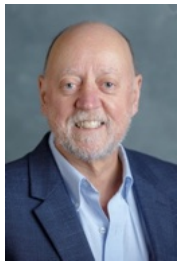


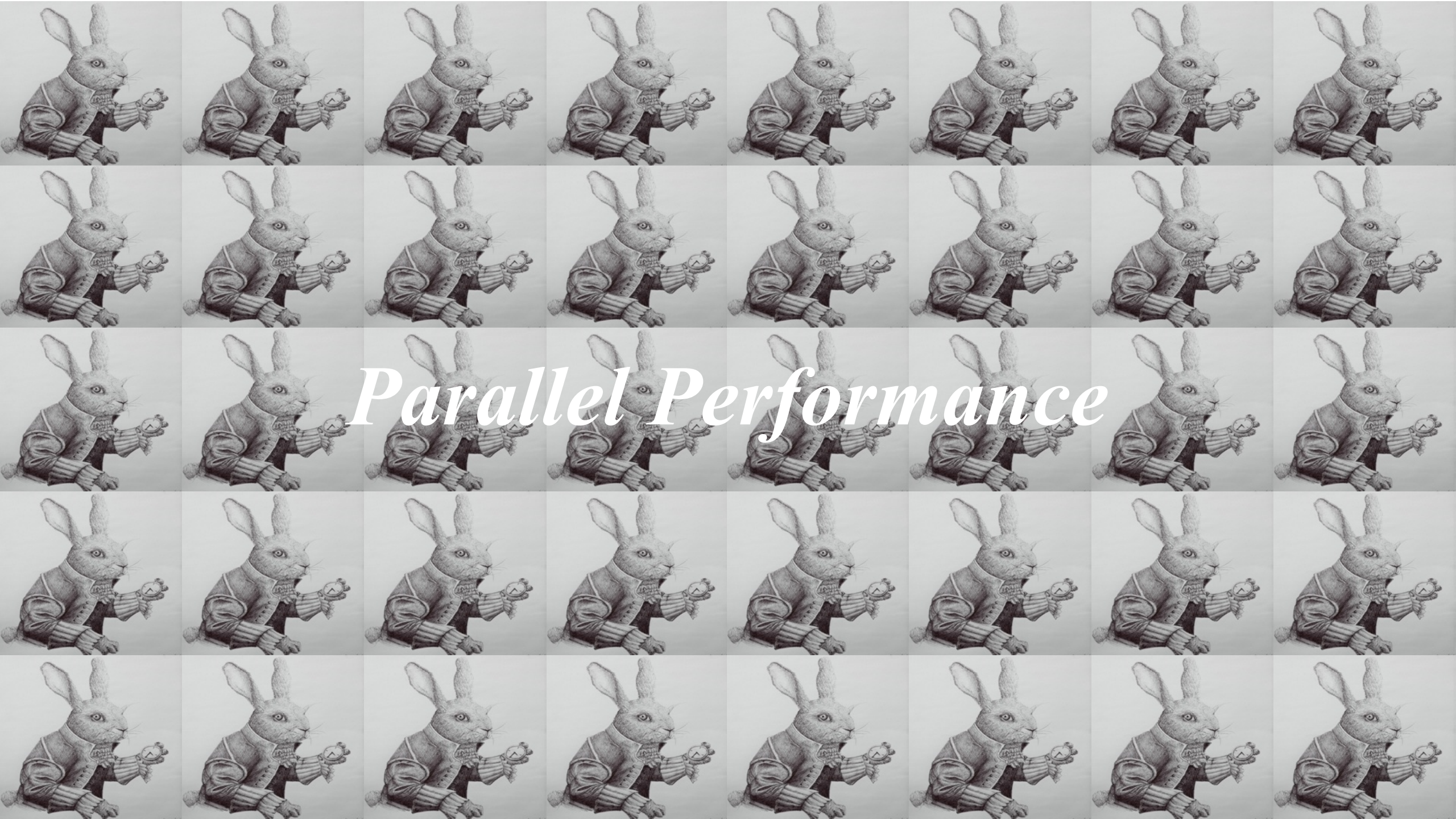
Parallel programming

Numerical libraries

Communication libraries

"A Not So Simple Matter of Software"
Jack Dongarra, ACM Turing Award,
Talk at SC 2022




Parallel Performance

A Not So Simple Matter of Performance

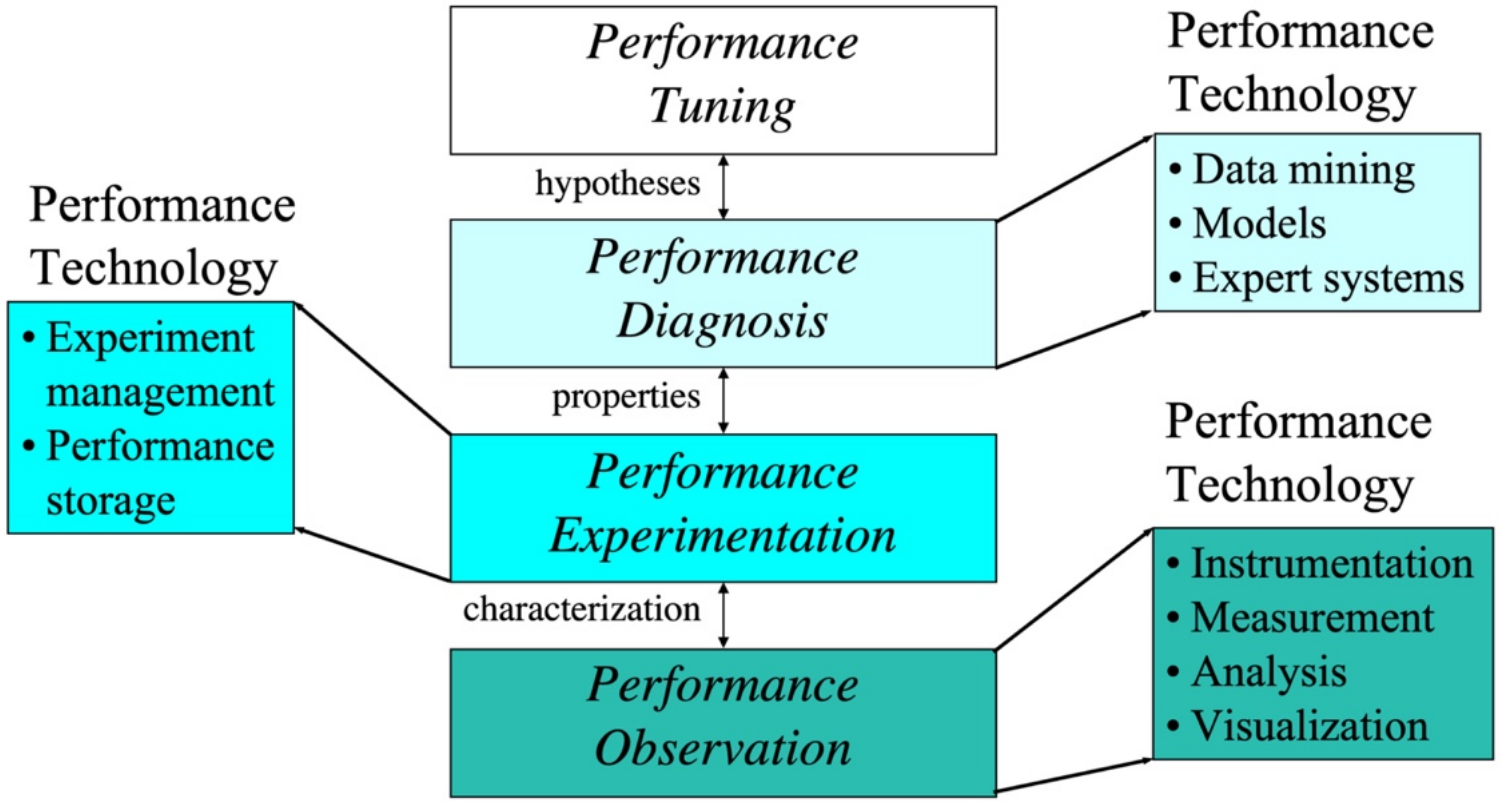
- I am interested in fundamental aspects of parallel performance
  - Understand the performance of a parallel computation on an HPC machine
- Understanding performance is a knowledge-based process
  - ***Observation***: measure and characterize performance behavior
  - ***Diagnosis***: identify and explain performance problems
  - ***Tuning***: recommend improve (optimize) performance
- How to make the process more effective and productive?
  - What are methods and best practices for performance problem solving?
  - What techniques and tools are needed for observability, analysis, and optimization
- How to build and integrate *performance technology*
  - To obtain, learn, and carry forward performance knowledge
  - To address increasing scalability and complexity issues
  - To incorporate performance awareness in execution models

Performance Observability and Uncertainty

- What is the "true" parallel computation performance?
- Performance observability is the basis of performance understanding
- Measurement Uncertainty Principle (see my Ph.D. thesis)
  - Any performance measurement will be *intrusive*
  - Performance intrusion might result in *perturbation*
  - Measurement degree is inversely proportional to intrusion and perturbation
  - Can we analyze and remove perturbation when it occurs? … Yes and No
- Performance analysis is based on performance measures
  - How is "accuracy" of performance analysis evaluated?
  - How is this done when "true" performance is unknown?
- What to do?
  - Develop robust *performance systems*
  - Rationalize performance measurement outcomes

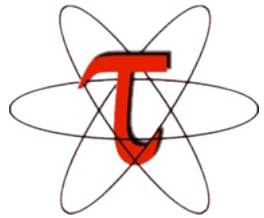
Parallel Performance Process

- Follow an empirically-based approach
- Performance technology developed for each level



Performance Technology Eras

- HPC performance technology has evolved to serve the dominant architectures and programming models
  - Observability era (1991 – 1998)
    - instrumentation, measurement, analysis, intrusion
  - Diagnosis era (1998 – 2007)
    - identifying performance inefficiencies, problem modeling
  - Complexity era (2008 – 2012)
    - scale, memory hierarchy, network, multicore, GPU
  - Productivity (Exascale) (2013 – future)
    - heterogeneity, richer applications, big data, ML, exascale
    - involves system-wide performance concerns

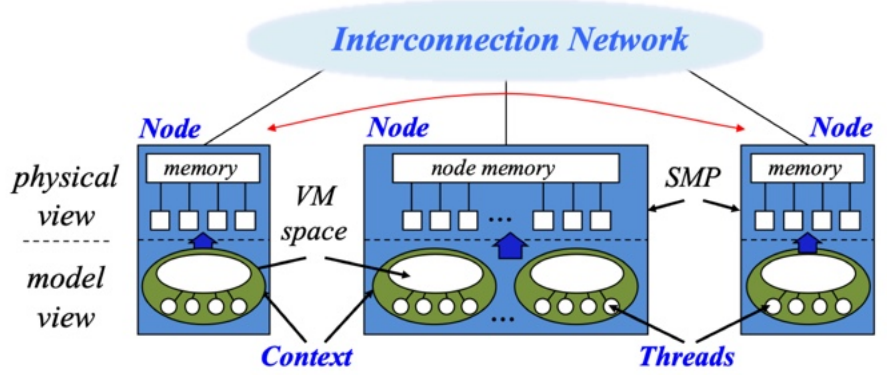


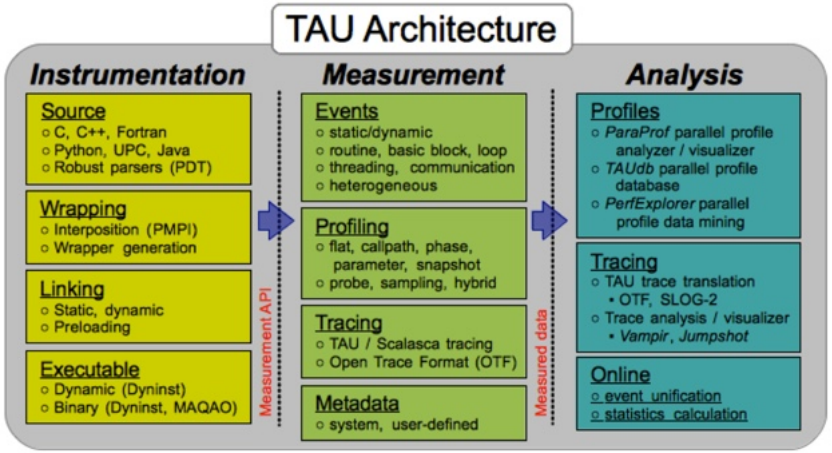
- Applications are now more complex and diverse as HPC extends to scientific workflows, big data analytics, AI computing

TAU Project

http://tau.uoregon.edu

- Research and development effort spanning 30+ years
- Focus on parallel performance problems and technologies
- *TAU Performance System®* research program
- Performance problem solving *framework* for HPC
  - Integrated, scalable, flexible, portable
  - Target all parallel programming / execution paradigms
- Integrated performance *toolsuite* (TAU)
  - Multi-level performance instrumentation
  - Flexible and configurable performance measurement
  - Widely-ported performance profiling / tracing system
  - Performance data management and data mining
  - Open source (BSD-style license)
- Broad use for performance analysis and engineering in complex software, systems, applications





TAU Project History

1992-1995: DARPA pC++ (Gannon, Malony, Mohr). TAU (Tools Are Us) is born. *[parallel profiling, tracing, performance extrapolation]*

1995-1998: Shende Ph.D. (performance mapping, instrumentation). TAU v1.0. *[multiple languages, source analysis, automatic instrumentation]*

Observability

1998-2001: Significant effort in Fortran analysis and instrumentation, work with Mohr on OpenMP, Kojak tracing integration, focus on automated performance analysis. *[performance diagnosis, source analysis, instrumentation]*

2002-2005: Focus on profiling analysis, measurement scalability, and perturbation compensation. *[analysis, scalability, perturbation analysis, applications]*

2005-2007: More emphasis on tool integration, usability, and data presentation. TAU v2.0 released. *[performance visualization, binary instrumentation, integration, performance diagnosis and modeling]*

Diagnosis

2008-2011: Add performance database support, data mining, and rule-based analysis. Develop measurement/analysis for heterogeneous systems. Core measurement infrastructure integration (Score-P). *[database, data mining, expert system, heterogeneous measurement, infrastructure integration]*

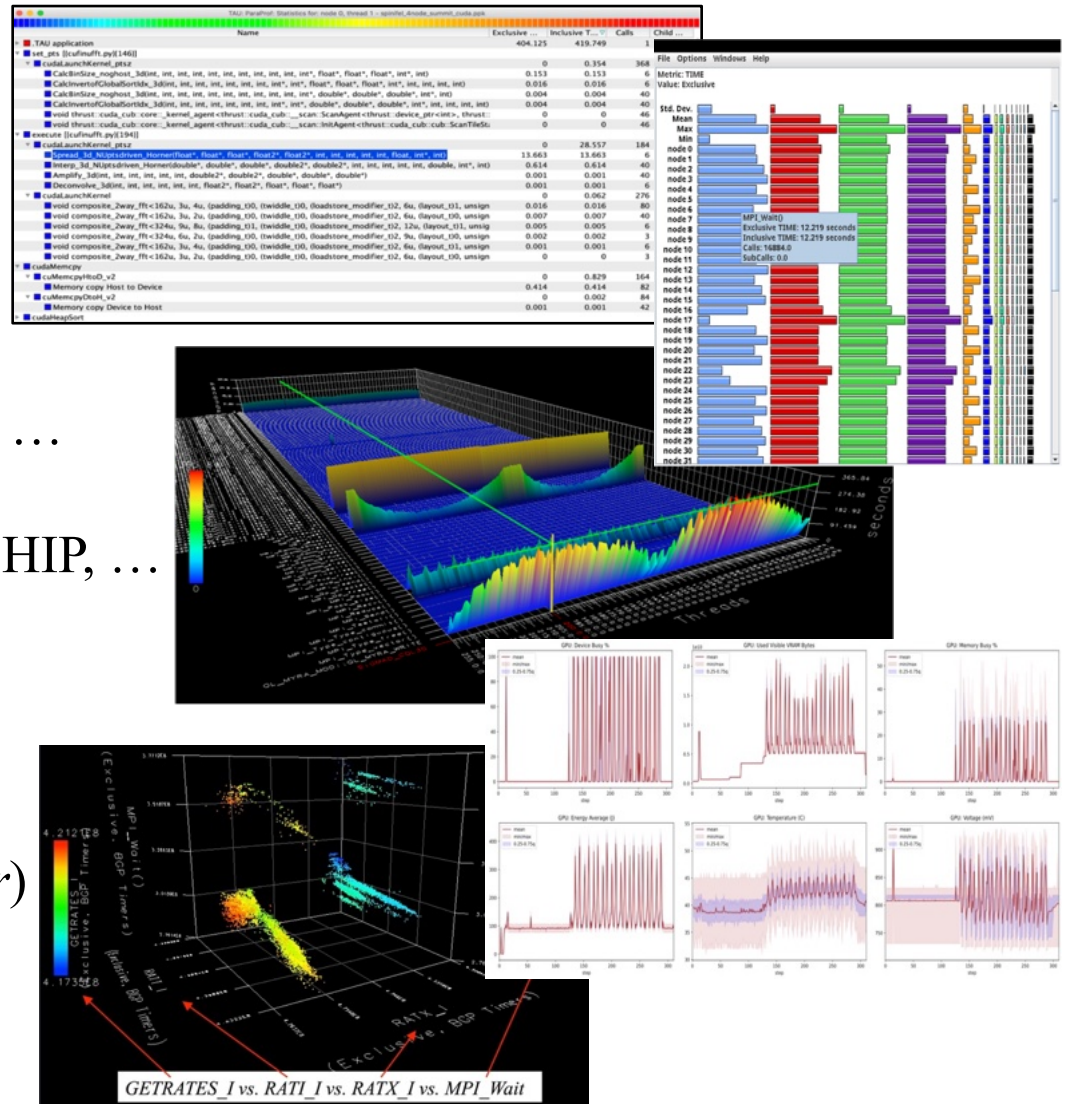
Complexity

2012-present: Focus on exascale systems. Improve scalability, heterogeneous support, runtime system integration, dynamic adaptation. Apply to petascale / exascale applications. *[scale, autotuning, user-level]*

Exascale

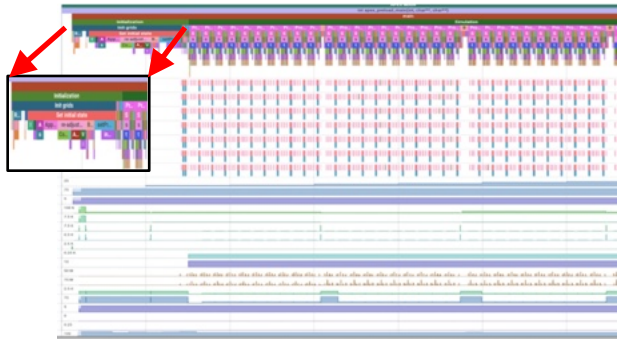
TAU Performance System (in a nutshell)

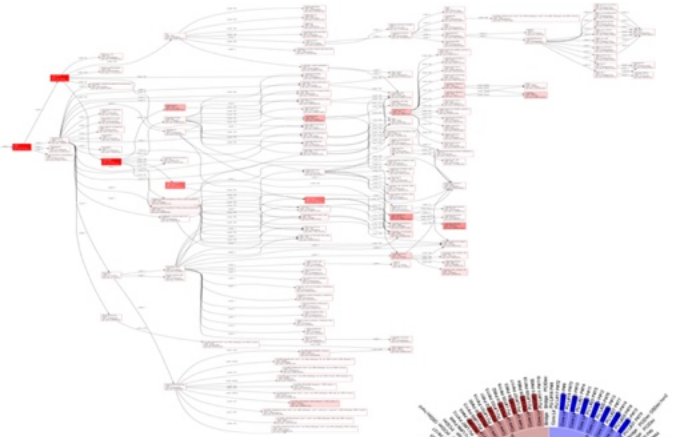
- Instrumentation
  - Fortran, C, C++, OpenMP, MPI, Python, Kokkos, …
  - Source, compiler, library wrapping, binary rewriting
  - Automatic instrumentation
- Measurement
  - Probe-based and sample-based supported
  - Internode: MPI, OpenSHMEM, ARMCI, PGAS, DMAPP, …
  - Intranode: Pthreads, OpenMP, OpenACC, hybrid, …
  - Heterogeneous: GPU, MIC, CUDA, OpenCL, OpenACC, HIP, …
  - Performance data (timing, counters) and metadata
  - Parallel profiling and tracing
- Analysis
  - Parallel profile analysis and visualization (*ParaProf*)
  - Performance data mining / machine learning (*PerfExplorer*)
  - Performance database technology (*TAUdb*)
  - Empirical autotuning

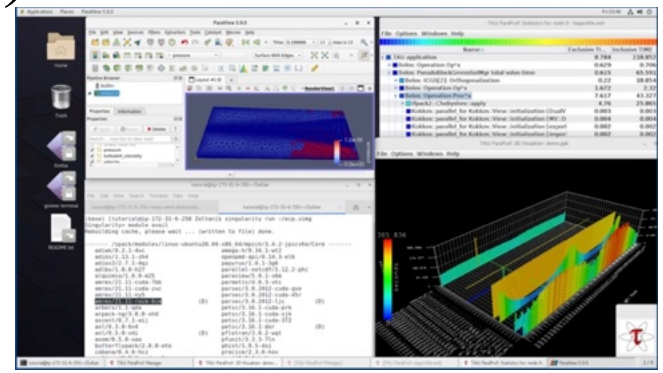


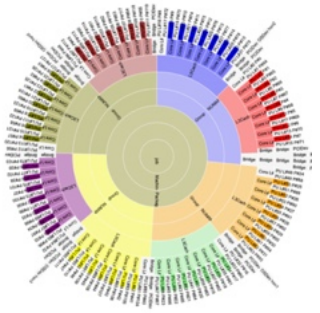
Recent Research Focus

- Heterogeneous systems performance technologies
  - Cross-platform accelerated computing (CPU+GPU)
  - Programming systems for performance portability
- Task-based performance analysis (*APEX*)
- LLVM-based code rewriting (*MARTINI*)
- Service-based in situ workflow analysis (*SOMA*)
- Exascale Scientific Software Stack (*E4S*)
- HPC resource monitoring (*ZeroSum*)
- HPC application monitoring (*SIMON*)



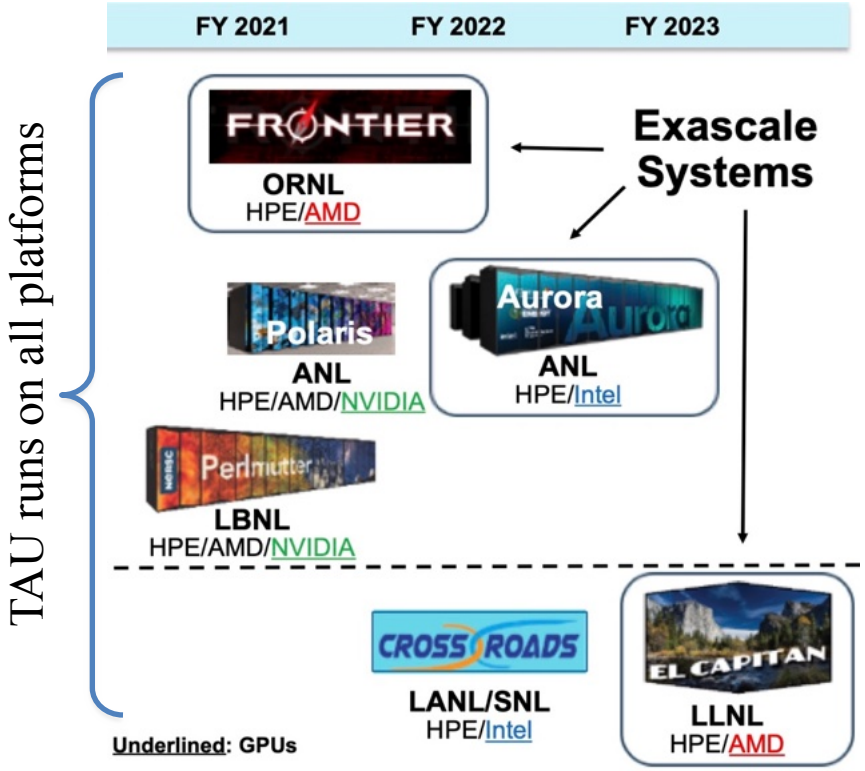


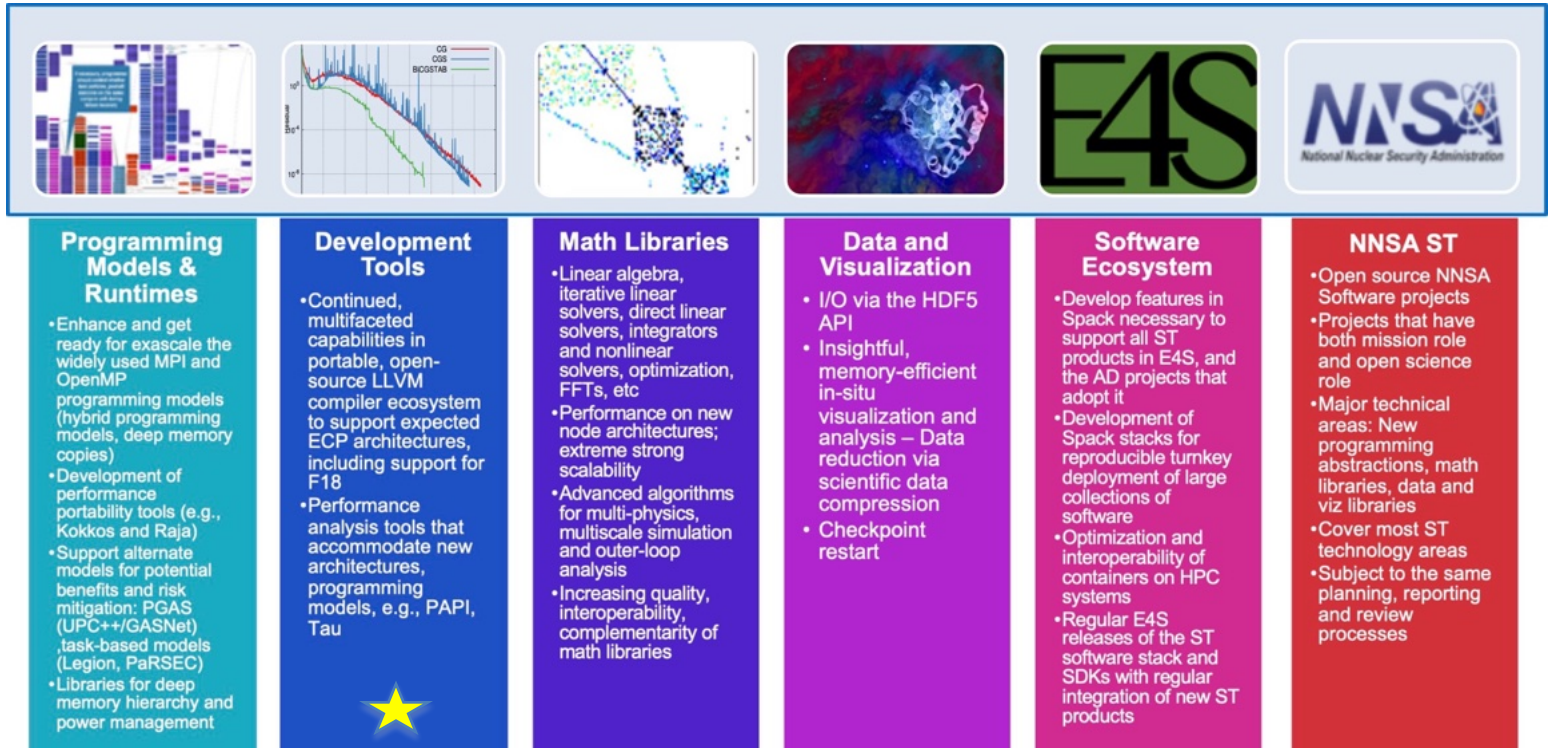




Exascale Computing Project (2016-2024)

- U.S. Department of Energy
- Develop exascale-ready applications of national interest
- Create scientific software stack on pre-exascale and exascale systems





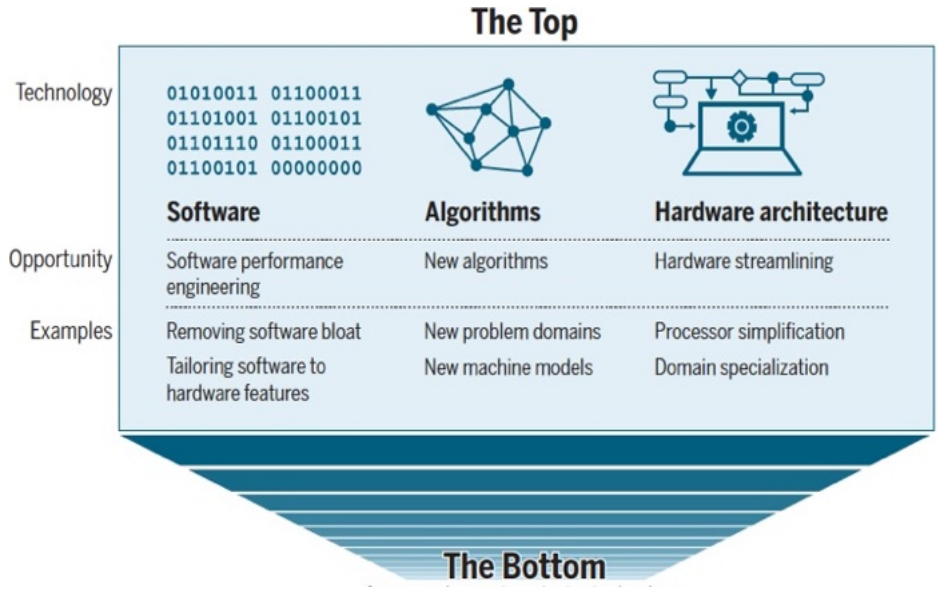

FUTURE

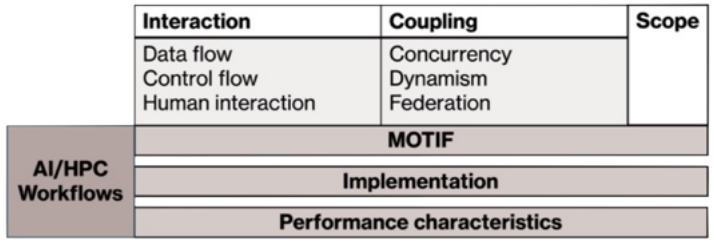
Future of Supercomputing

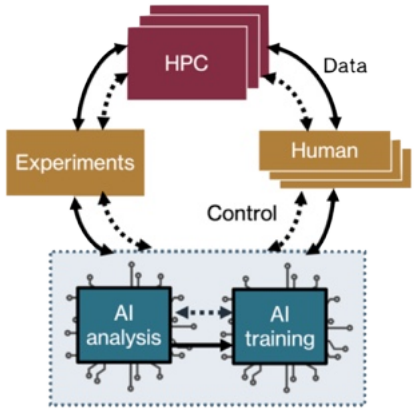
- Dynamics in the HPC field
- Concern for the end of Moore's Law
- Large-scale HPC systems for scientific computing built and proposed
  - EuroHPC JUPITER (exascale)
  - Japan's FugakuNEXT (zettascale, 2030)
- Expanding influence of AI-computing
  - Modeling and Simulation
  - AI-coupled HPC workflows
  - Foundation models

"There's plenty of room at the Top: What will drive computer performance after Moore's law?"

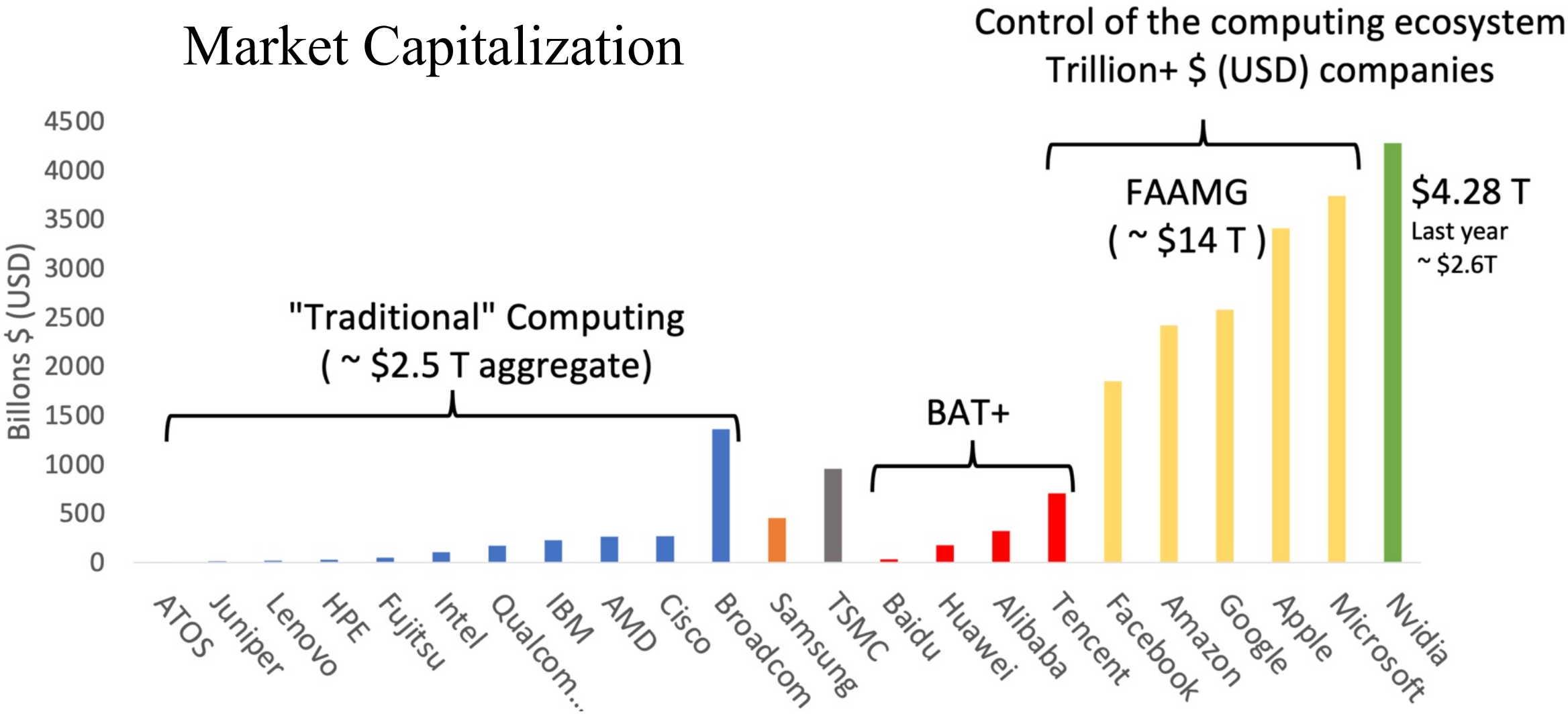
Leiserson *et al.*, *Science* **368**, 1079 (2020) 5 June 2020





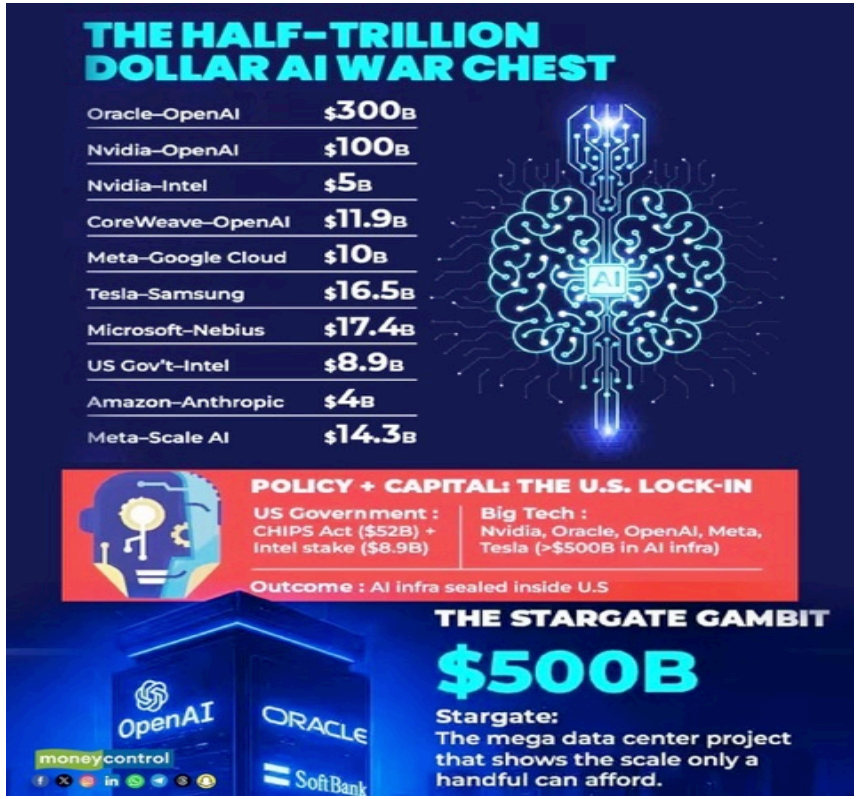


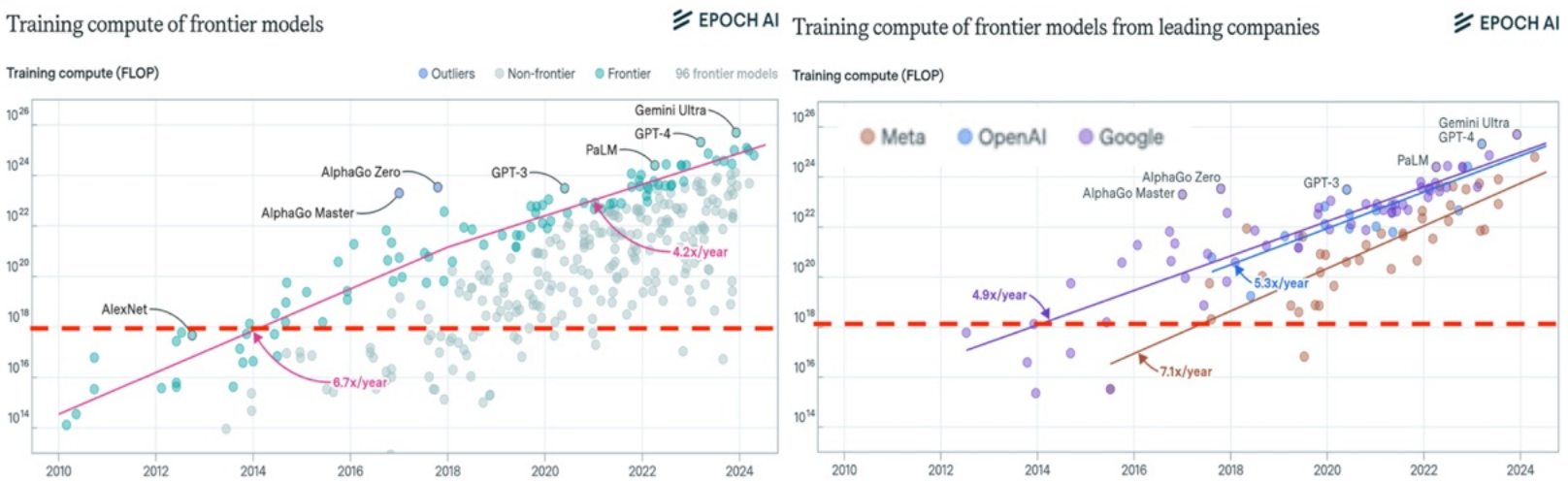
Measure of Market Influence (2025)



Rise of AI

- AI training and inference driving HPC technology industry
- Foundation and frontier models
- Huge computational and data requirements
- Huge power requirements
- Huge investments





El Capitan	$600M	30MW
Frontier	$600M	25MW
Aurora	$500M	39MW
JUPITER	€500M	13MW
FugakuNEXT	¥110B	??
xAI Colossus	$3-4B	300MW

Take Away Thoughts

- HPC is about parallelism and performance
- Each has fundamental computer science aspects
- Supercomputing is about scale
- What drives supercomputing systems
  - Current problems are important to address
  - Business markets
- Supercomputing concerns
  - Efficiency: performance and power
  - AI-coupled HPC might play an important role here

- "The future is not what it used to be ." Yogi Berra

Special Surprise

- ***SC*** is the International Conference for High Performance Computing, Networking, Storage, and Analysis
  - Established in 1988 (https://supercomputing.org/conference-history/)
  - Association for Computing Machinery and IEEE Computer Society
- I have been to every SC conference (known as an SC Perennial!)
- One year we made TAU pins to give away (way to many)
- You are welcome to take one