50 YEARS OF SUPERCOMPUTING

The Evolution of a Powerful Tool

- a personal perspective -

Argonne Training Program on Extreme-Scale Computing (ATPESC 2023)
August 9, 2023

Argonne NATIONAL LABORATORY

Brief History of Supercomputing

CHAPMAN & HALL/CRC COMPUTATIONAL SCIENCE SERIES
Unmatched
50 Years of Supercomputing
David Barkai
CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Big Iron Vector Processors	Multiprocessors	Microprocessors	Clusters	Accelerators Cloud
'70s	'80s	'90s	'00s	'10s

Why “Unmatched”?

Moore’s Law sets the bar for rate of improved performance over time – 2x every 2 years

(yes, it’s about semiconductor feature density, and the compute speedup is at a slower rate as of late)

Top HPC systems: A factor of 1,000x every ~ 12 years

1st GF: XMP @ 0.8GF ‘82, Cray-2 @ 1.9GF ’85

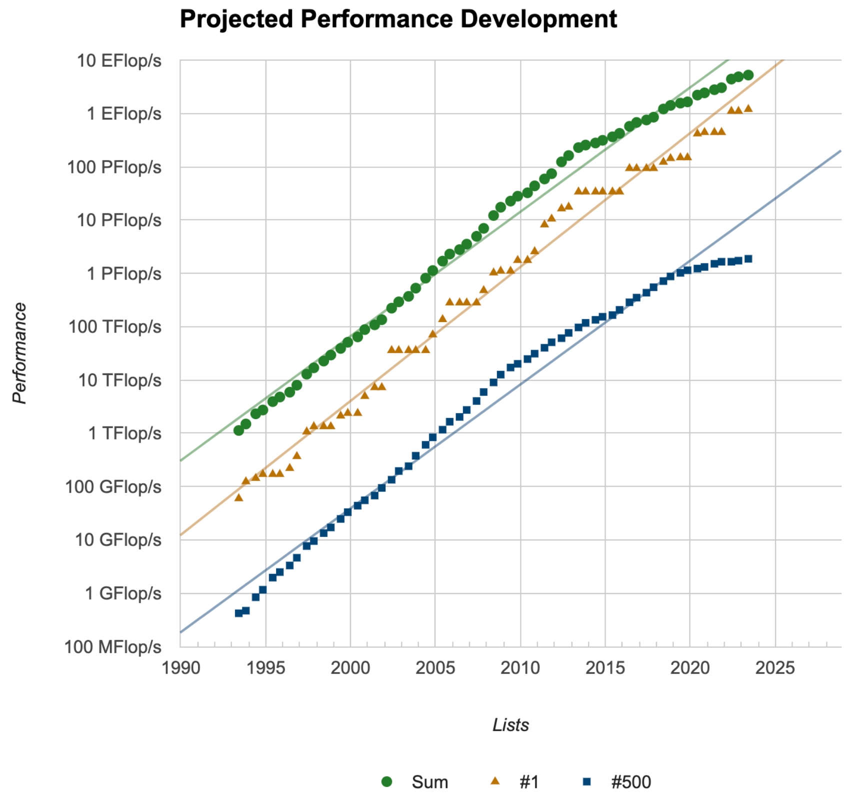
1st TF: ’96 (ASCI Red)

1st PF: ’08 (Road Runner)

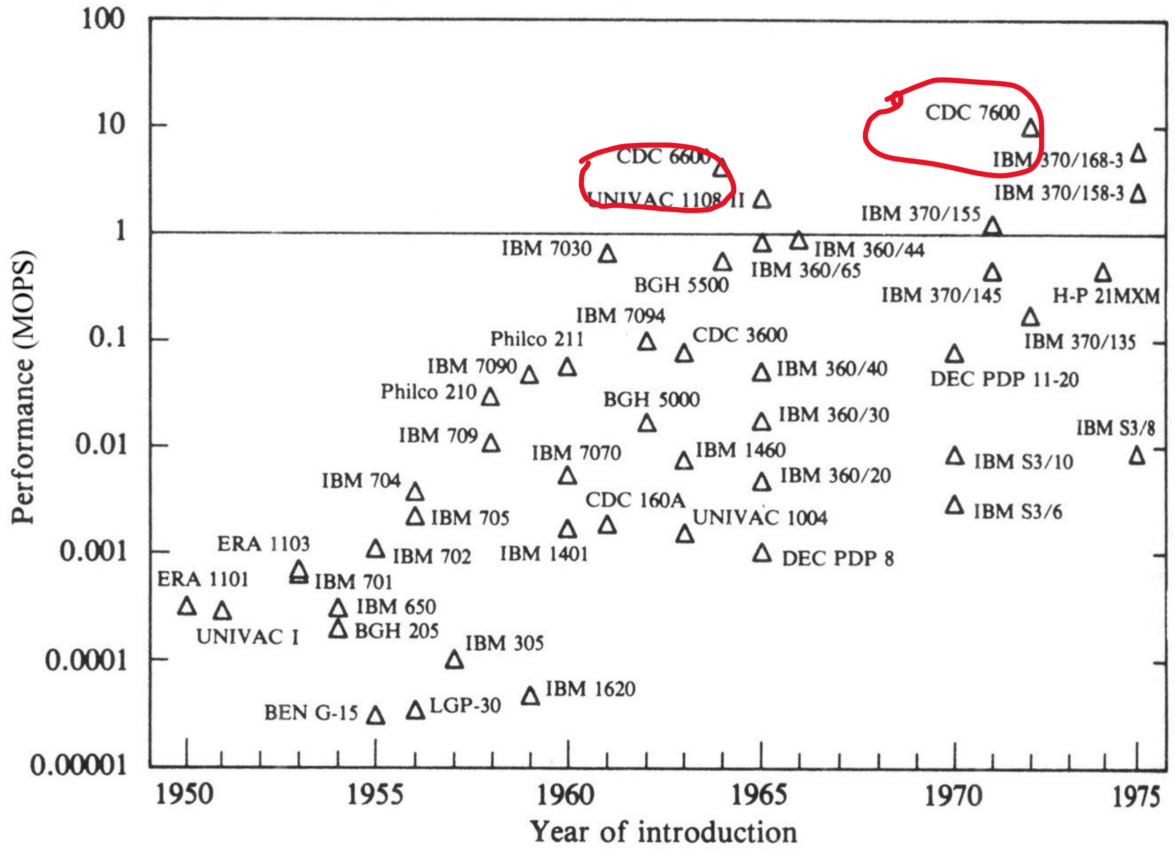
1st EF: ‘22 (Frontier)

In 12 years the fastest moving ”unit” technology advances by 64x

Supercomputing performance has been growing at a rate 15x higher than that of Moore’s Law



Early Days

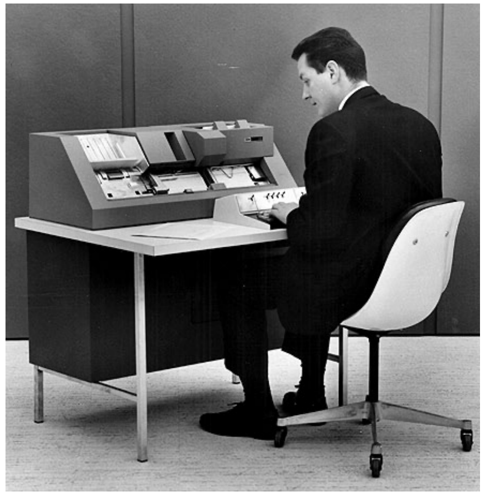


Source: Kenneth Flamm, "Creating the Computer", 1988.

1M ops/sec barrier passed ~1963

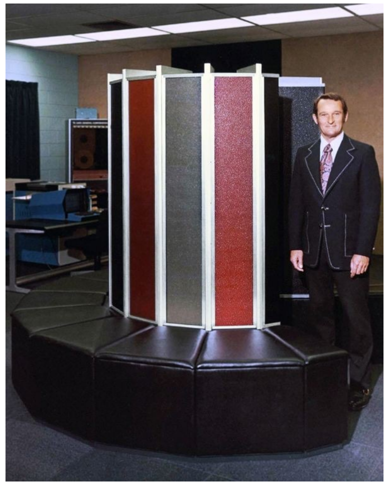
CDC-7600, at 36 MFlops, the most powerful in pre-vector era

Source: IT History Society



Source: IBM Archives

Vector Processors of the '70s



CDC Cyber 205

The ideas:

- Avoid wait time, *latency*, of sequential operations
- Use functional units *concurrently*
- Non-conflicted, *independent,* operations can overlap – foundation of parallelism
- *Arrays* are common; design instructions for arrays
- Multi-cycle operations to be *pipelines*.

The Cray-1 (1976) was not the first vector processor
- Texas Instrument's ASC and CDC's STAR-100 were delivered in 1973; it was the 1st commercially successful.

Competing approaches: memory-to-memory vs. vector registers
- the latter won.

CDC: APL instructions, native 32-bit double result rate
Cray: Vector registers, chaining

The providers of high-end computing were sometimes referred to as "IBM and the seven dwarfs", the dwarfs being CDC, Honeywell, Univac, GE, RCA, Burroughs, NCR. Sometimes as the "BUNCH".. (dropping GE, RCA)

The '80s: Multiprocessors / Macro-Parallelism

Early MPs

Macro parallelism
Methodologies: Task or Data
Issues with Task approach: Identification, load balance, data shared, scaling

Attached Processors

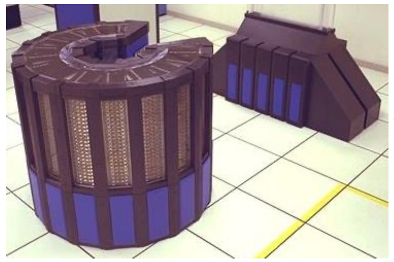
Mini-supers: Miniaturizing supers many vendors, short-lived impact

CDC 6500 – built 1967
Now in Living Computers Museum, Seattle
2 6500s ran 4P weather model at Fleet Numerical in 1970!

Mockup of Cray's 4-cpu CDC 8600 - never completed

ILLIAC IV – built 1970

Cray X-MP. Architected by Steve Chen. Followed by the Y-MP.



Cray-2. Decisively broke the GF barrier.

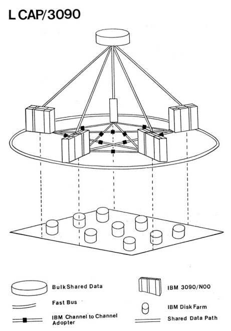


Fig. 3. ICAP-3 configuration. The IBM 3090/N00 mainframes are either IBM-3090/600 or IBM-3090/400.

ICAP: late 80s. 4 IBM 3090-400. 16-way, 4 distributed memory blocks. And 10-20 FPS 264s attached processors.

The ‘90s: Microprocessors / Massive Parallelism

MPPs from the late ‘80s – experimental and commercial
Unlike the minis, built from small microprocessors
Foundational for clusters: Distributed Memory, Network

Goodyear MPP. 16K 1-bit CPUs
SIMD

Thinking Machines’ CM-5. 1K SPARC CPUs
SIMD. Fat Tree.

In the first Top500 list, in1993, the CM-5 occupied the top 4 slots, with a 5th one in the top 10.

Intel enters HPC as a system house
Justin Rattner: Paragon; hypercube
32-bit x86 chips for HPC

Intel Paragon XP-S at ORNL, ‘94

Large-scale parallelism on distributed memory systems

ASCI Red, Intel-Sandia ‘partitioned’ design. x86 CPUs.
First Teraflops system on Top500 (‘96)

The '00s: Clusters / Commodity / Standards

Starting in the '90s the transition from the old *silo culture* to *standards* and *open source* is now as complete as it probably can get
(some proprietary networks, compilers, libraries)

Big Iron vector processors lose steam (~5-year dev cycle)

Fine and Coarse parallelism learnt, and tools established

Commodity microprocessors, 32-bit Fl. Pt., within O(1)-O(2) of vec. proc.

Cluster Proof-of-Concept in the '90s: Beowulf, NOW, others

New ecosystem / business model emerges:

Technology Providers --- System Houses --- Community Software

Typical cluster:
Two-CPU nodes stacked in 'standard' cabinets/racks, all connected via a fat-tree network, running Linux, programmed in Fortran, C, C++, supported by MPI

Differentiation becomes more subtle

Accelerators make an entrance – the new attached processors

A Beowulf cluster with Thomas Sterling

By the late '80s in Cray Research they realized that software development cost is as high as that for its hardware. Switched OS to Unix derivative

The ‘10s: Accelerators / Cloud Computing

The Changing Face of HPC

The Application Space:

“Classic HPC” – numerical computations for simulations of physical systems and other mathematical problems

Data Analytics – ever larger datasets require high-end systems

Machine Learning – applied to those large datasets require same high-end systems

DA and ML/AI applications embraced by, and fall under, the HPC umbrella

Classic HPC applications make increasingly greater use of DA and ML methods (size of data and complexity)

Delivery of Cycles:

Accelerators are becoming the main source of compute cycles

Cloud computing is increasingly supplements, and often replaces, on-premise computer centers

Processors: x86, GPU, but also ARM, and QC

A Tribute to Japan’s Supercomputers – The Amazing 9-Year Cadence

The repeated successes of government-academia-enterprise collaboration

One-off innovative designs that captured the #1 spot

Numerical Wind Tunnel — ‘93

Earth Simulator — ‘02

K — ‘11

Fugaku — ‘20

?? — ‘29

Beyond and Behind the HPC's Evolution

Planning Ahead and Codesign

Expressing Parallelism

Algorithms vs Hardware

Features that come back

RISC vs CISC

Lessons from Screwups

Benefits from HPC

<u>Another time …</u>

Tracking Applications over time

On Performance

On Productivity

On Standards

On Programming Languages

HPC apps for Society

Planning and Codesign – First Step

A remarkable '94 3-day workshop – *"Enabling Technologies for Petaflops Computing"*

Teams: Applications, Devices, Systems, Software

Predictions / Outcomes (based on Roadrunner and Jaguar):

PF in 2014 / 2008
CPU descendent of mid-90s MPs / missed
Memory << 1byte/flop / partially right, but ~10x bigger mem
Radical departure from data access methods / wrong. MPI it is
Mass-volume market determines rate of progress / Correct
Semiconductor technology; optical for inter-proc. / Correct (little optical)
Part count: 100K to 1M / Correct when including memory, network
Power ~1MW / >2x for RR; 7MW for Jaguar

"Server" not uttered. x86 for PF not foreseen.
Biggest "miss": Role of accelerators (today's GPUs)

The Systems and Architecture group at the Pasadena workshop

Roadrunner at LANL

Jaguar at LANL

Seymour predicted: Teraflops system in '98 at a cost of $50M
ASCI Red: Mid-'97, for $46M
(wrong about type pf processor)

Planning and Codesign – IESP

The “*International Exascale Software Project*”

Driven and managed by the DOE, international in scope with Government (users), Academia (research), and Industry (technology, vendors) participation. A series of workshops starting in 2009 and spanning a couple of years, with numerous projects spawning from it.

Mission: Prepare for exascale high-resolution data-intensive an open-source common computational environment: X-stack.

Most quoted product – The IESP Roadmap. Scoring:

Timing: EF in ‘20 (yes with Fugaku’s mixed precision HPL). Frontier 2 years later.
Node: Unexpected massive role of GPUs (Fugaku with ARM and no accelerator is an exception)
Resulting in ~10K nodes (not O(100K)) simpler system
The power constraint of 20MW/EF is satisfied by Frontier
Software overhaul, top to bottom, not needed (as feared by some)

Codesign applies mostly to software. Hardware codesign is harder..
Different paths for each geo: x86+GPU ‘fat node’ cluster (US),
ARM with vector (Japan), RISC-V (Europe), own development (China)

Perspective on how far we’ve come:
A single Frontier node’s peak is 150TF.
76 Cabinets of ASCI Red add up to 1 TF
Node to node comparison: ~700,000x perf increase over 25 years

On Expressing Parallelism

Vector → MPs → Massive parallelism → multicore → Threads → vectors in core → GPUs

Vectorization: Explicit → Code 'assistance' to compiler → Directives → Auto-vectorize (always disappointing)

MPs: Macro/coarse para + vector

Clusters: [on-chip + on-node + attached accelerators] x many nodes

Complexity: multi-layers, distributed memory on top of shared blocks
Language/Compiler vs, Library Solution:

Shared memory parallelism conceded to *Language* + *Directive* (OpenMP etc.)

Dist. Mem para is harder to express due to lack of global address space

Coarrays is a language solution for Fortran. MPI became the norm as a library solution – and used almost exclusively

Coarrays conceived '91
Elegant solution, but >15 years to Standard
Language-specific; hard to revise

←

Why 'Library' won
But split tools
No unified model

→

MPI conceived '92, released '94
Community driven, language-independent, amenable to experimentation and quick update cycles

Algorithms vs. Hardware

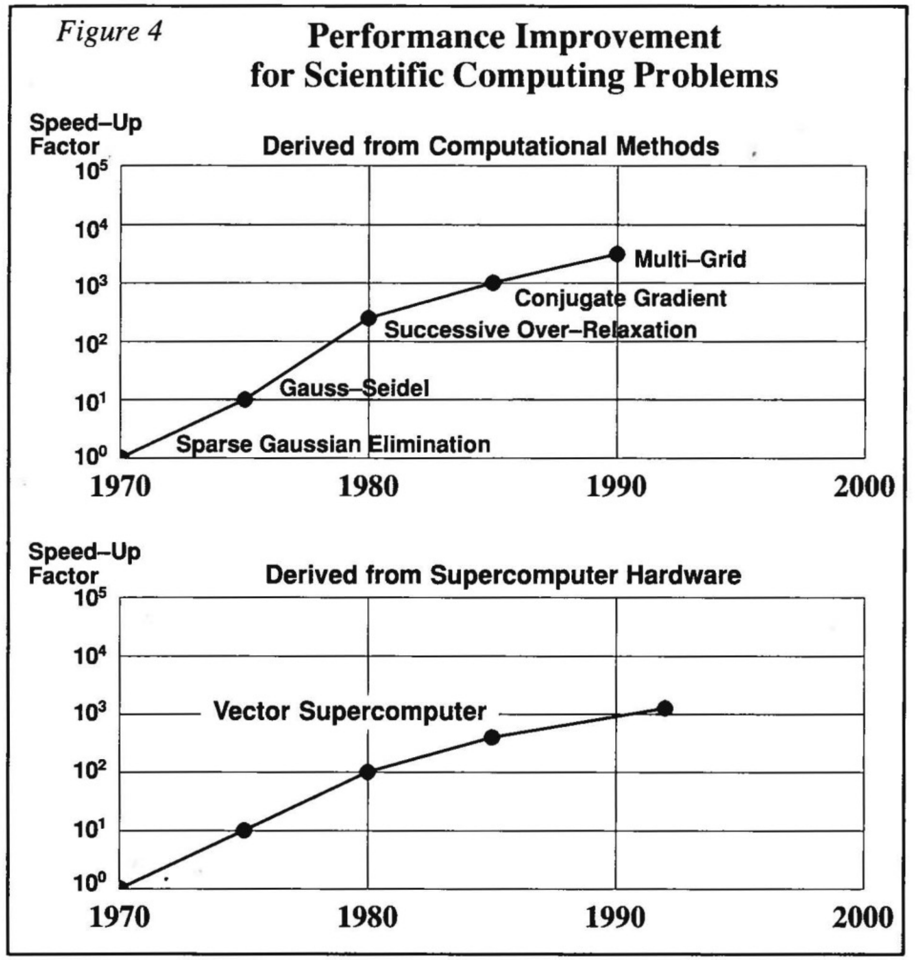
'92 "Blue Book" study by High Performance Computing and Communications Working Group reporting to the Committee on Physical, Mathematical, and Engineering Sciences of the Federal Coordinating Council for Science, Engineering, and Technology

Addressing 'Grand Challenges'

The claim: Advances in hardware and in computational methods each contributed speed-ups of 1,000 times over 20-year period

The message "algorithms contribute significantly to speed-up" is, no doubt, correct.

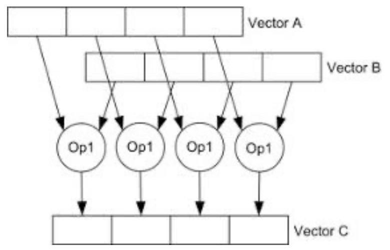
- The hardware speed-up is somewhat exaggerated
- The report offers no details and data to justify the algorithm chart
- Estimates likely done through the solver's ops and iterations count; this is largely irrelevant in today's parallel systems



Features That Come Back

Innovation includes discovering new applications and new implementations of old ideas

Vector Instructions
Mostly disappeared in the '90s, to reappear 10-15 years later – but played a lesser role in achieving performance
Left behind awareness of independent computations and arrays
Code 'organized' for vector execution often runs faster sequentially too



Attached Processors
From signal processing to HPC in the '80s (FPS)
Absent in the '90s
Now in full strengths as GPUs

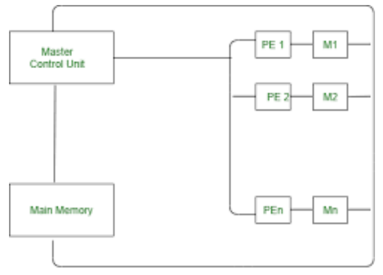
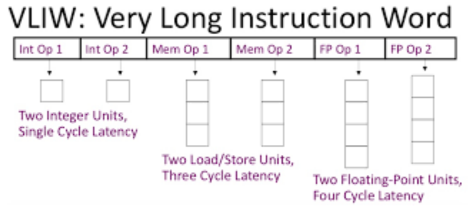


Figure - SIMD Array Processor Organization

VLIW Architecture
Great in theory: pipelines, multiple functional units, parallel ops
'80s designs (e.g., Multiflow)
Fades away, then Itanium (a failure)
Resurrected in GPUs


VLIW: Very Long Instruction Word
Int Op 1
Int Op 2
Mem Op 1
Mem Op 2
FP Op 1
FP Op 2
Two Integer Units, Single Cycle Latency
Two Load/Store Units, Three Cycle Latency
Two Floating-Point Units, Four Cycle Latency

Pop Count Instruction
Returns # of 1's in a word
From the '60s at the NSA's behest – for cryptanalysis
Resurrected in modern microprocessors as aid for
New uses:
Error Correction, Neural networks, Chess, Compiler opt

RISC vs. CISC

Complex or Reduced Instruction Set Computer – Why it matters

CISC is what comes out if a user decides, without knowledge or consideration of how their computer works
- e.g., one instruction for "add 2 numbers; tell the answer"

RISC is what comes out when a computer architect listens to the user, then break it up to the operations that correspond to the computer's components: "load A, load B, add A and B, Store result"

CISC: fewer instructions, multi-cycle per
RISC: more, but shorter (both in time and format), instructions

Why did RISC prevail for HPC?
Simpler logic to implement; less gates
Makes it possible to overlap, pipeline, and express parallelism
(CISC is suitable for appliance-type devices)

What about x86?
Legacy code and backwards compatibility force keeping it CISC
Chip hardware is RISC with microcode layer between CISC binary and hardware
(FDIV bug showed Intel value of microcode)

VLIW?
Makes sense only with RISC

Embracing Failure: Some ‘Screwups’ and Lessons

Floating Point Systems

FLOATING POINT SYSTEMS, INC.

Mid ‘80s, riding high with 64-bit attached processors
Betting the company on esoteric product (T-Series)
Took the company down. Causes:
Idealized app parameters; no ecosystem; design by ‘consultant’; Suppressed internal critique.

Intel Itanium

New architecture. Ecosystem needs to be built.
Cannot rely on “software will fix it”
Marketing hype goes only so far
Unintended consequences (late 64-bit Xeon, careers derailed, ..)

Supercomputer Systems Inc.

Spin-off Cray
Well funded by IBM; supported by US Gov.
Push technology boundaries
Big iron proprietary vector architecture “sell by” time has passed

Intel Xeon Phi

Research projects with shifting target use
Aggressive marketing
Heavy software investment
x86 advantage, but potential not realized (data feed)
Replaced by Intel GPU

Benefits from HPC

HPC impacts, if not always directly, all aspects of society
'91 GAO report highlights oil, automotive, aerospace, chemical, pharma
There's much more. Consider benefits from weather/climate modeling to sever weather alerts, agriculture, transportation, travel safety, water management, and more.

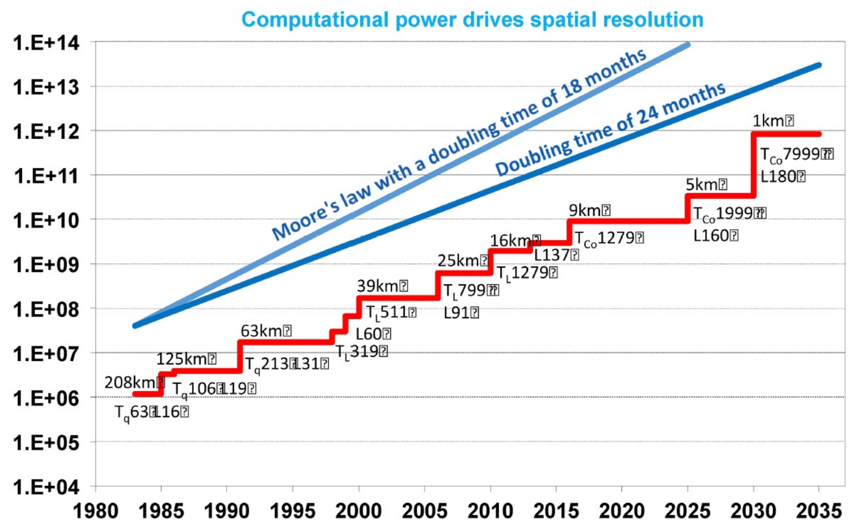
'Digital Twin' in design of engines, and aircraft, life sciences, earth system models

Quantifying economic returns is harder
IDC modeled HPC ROI (2013, 2017):

- ROI: $356 / $1 invested in HPC
- Profit: $38 / $1 invested
- Positive returns in under 2 years
- On average, innovation requires $3M
- Jobs created: ~30/site at cost of $93/job

Anecdotal examples:

- Model hepatitis C virus – savings $9B/year
- HPC for cancer clinical trials improve success rates, save time and costs
- Heart simulations at cell level reduce mortality
- Jet engine simulation improves efficiency; 1% translates to $2B annually
- Engines designed for biofuels to save over $1B/year
- Disaster mitigation saves lives and property (e.g., location and time-accurate landfall prediction of hurricanes saves lives and $100Ms)
- Decades of cost and time savings for automotive and aero



Resolution of ECMWF weather models tracks Moore's Law