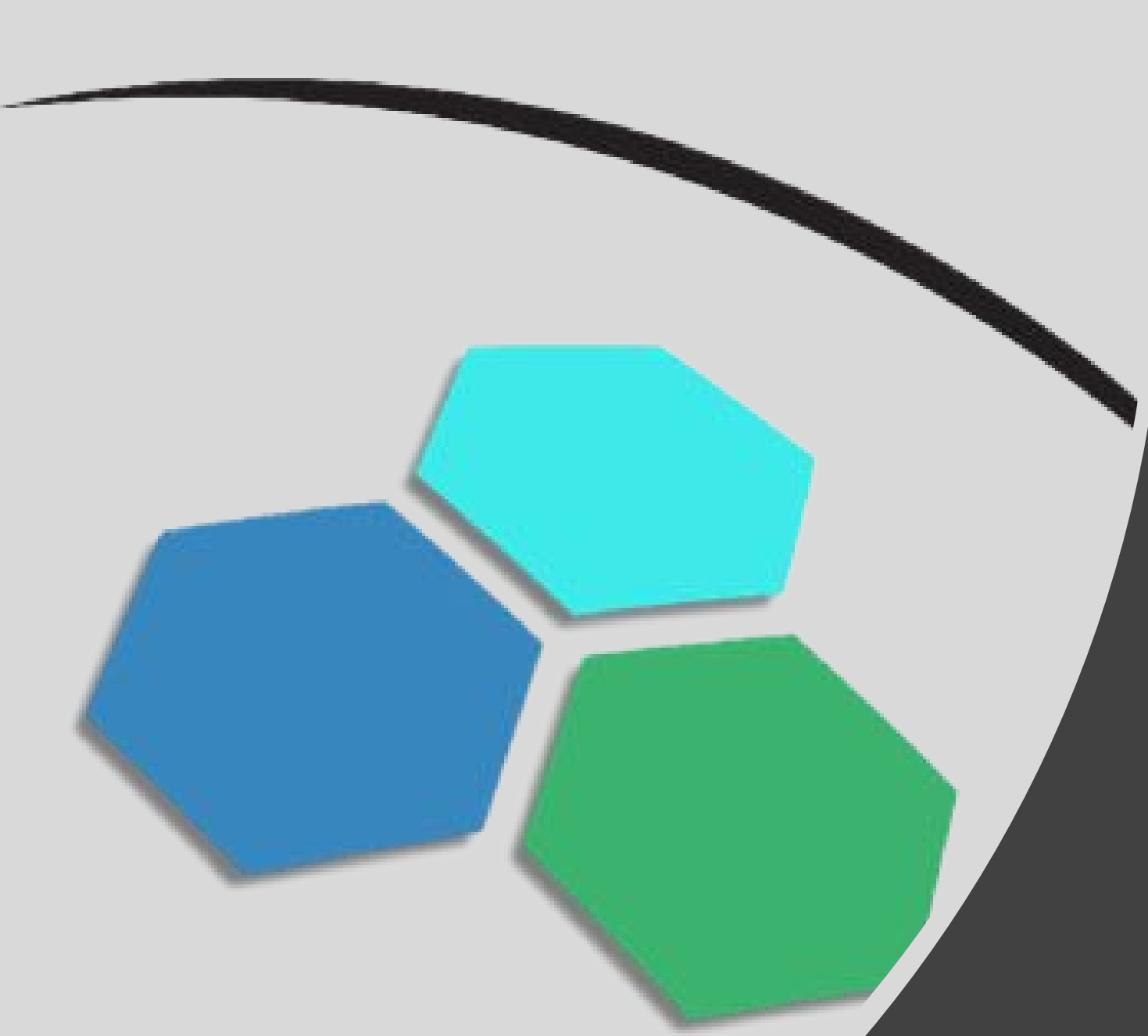


High Performance Computing: The Challenge of Computational and Data Science

Richard Loft
AreandDee, LLC
richard.d.loft@areanddee.com

October 29, 2024

AreandDee LLC
Come scale away...

COLORADO STATE UNIVERSITY

Outline

- Historical Background
- A tour inside of a Modern Supercomputer
- Recent NCAR Supercomputers
- New Paradigms: Computational and Data Science
- Parallel Computing
- Practical HPC: 10 Essentials for Successfully Using HPC Resources
- Q&A

What is a supercomputer?

- Any of a class of computing systems that are the fastest in terms of speed of calculation available at any given time, and generally (but not exclusively) used for scientific purposes.
- It's a marketing term! The term "Super Computing" was invented by IBM for the New York Worlds Fair in 1929.
- But the history of computing goes much farther back than that….

Analog Computers

Antikythera Mechanism
Inventor: Greek scientists (?)
Date: before 60 BCE
Purpose: Prediction of astronomical positions and events.

- Mechanical->Hydrolic-> Electrical
- Special Purpose
- Prediction of Phenomena

Tide predictor
Inventor: Lord Kelvin
Date: 1876
Purpose: Tide prediction by superposition of 10 tidal components (1876).

The First General Purpose Computer (1830s)

Ada Lovelace

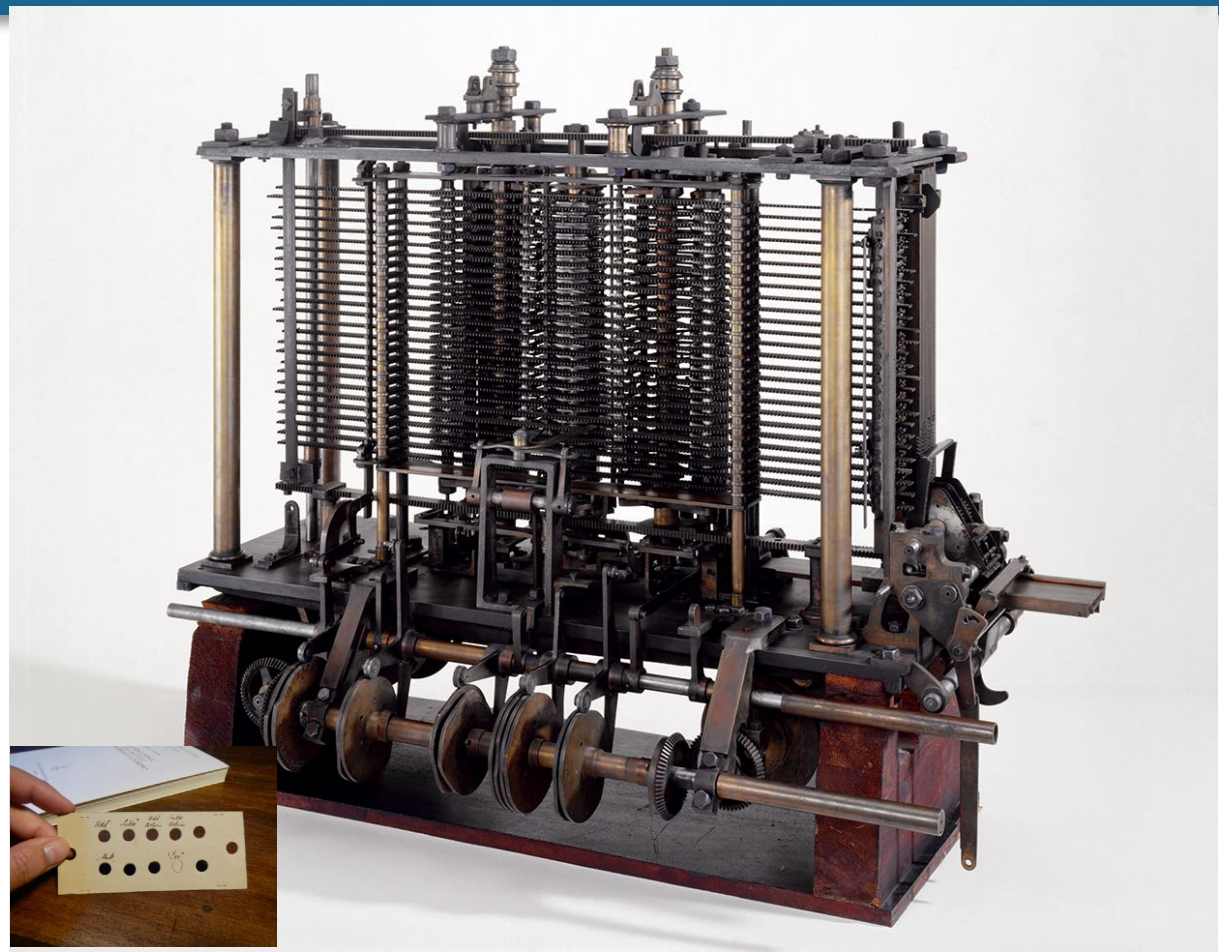
Charles Babbage

"…the Analytical Engine weaves algebraical patterns just as the Jacquard-loom weaves flowers and leaves".

Ada Lovelace

"As soon as an Analytical Engine exists, it will necessarily guide the future course of science."

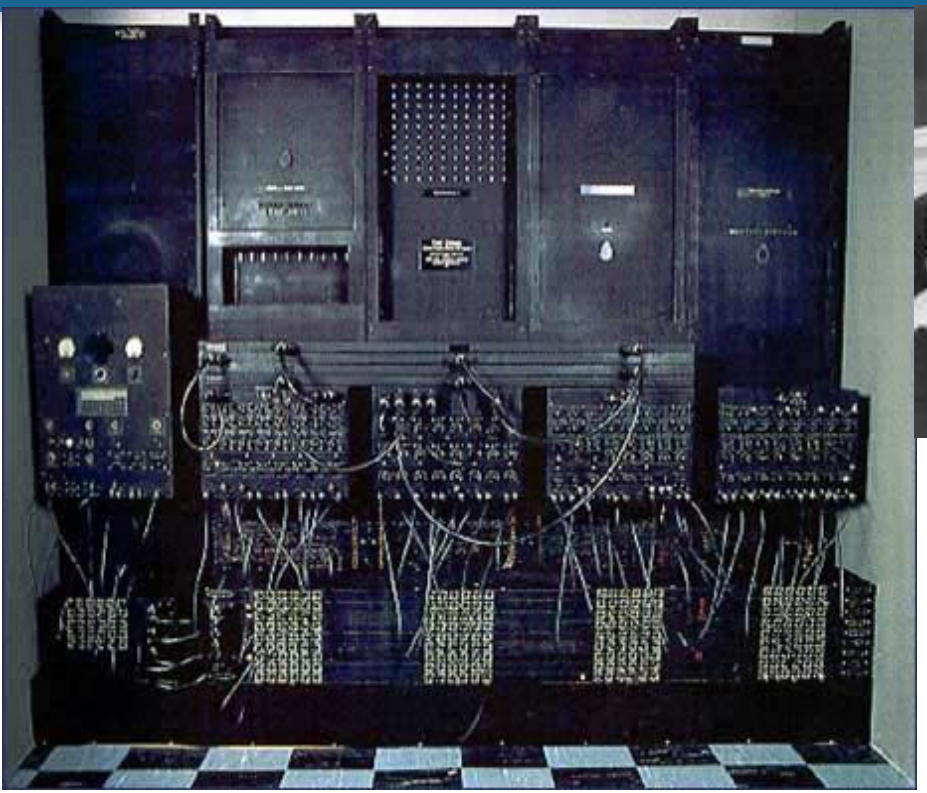
Charles Babbage



Punch Card

Babbage's Analytical Engine

ENIAC: Electronic Numerical Integrator and Computer



Inventors:
Mauchly & Eckert

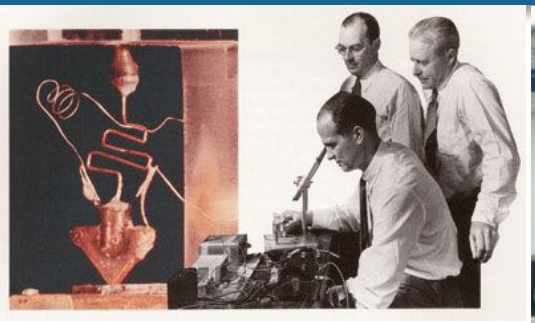
1950: The first NWP performed on ENIAC.

- Von Neumann
- Charney
- Fjörtoft

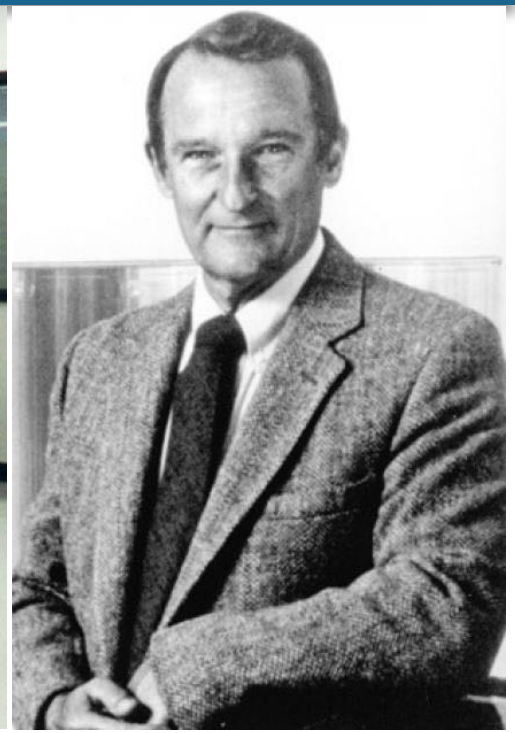
ENIAC (1945):
The first programmable, electronic, general-purpose digital computer.
Performance: 5000 additions/sec

Programmers:
- Betty Holberton,
- Jean Jennings Bartik,
- Kay McNulty,
- Marlyn Wescoff Meltzer,
- Ruth Lichterman, and
- Frances Bilas Spence

The Modern Semiconductor Era Begins

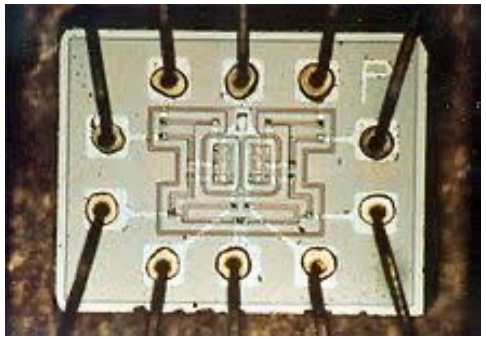


1947 - the transistor
Discovered at Bell Labs by William Shockley, John Bardeen and Walter Brattain



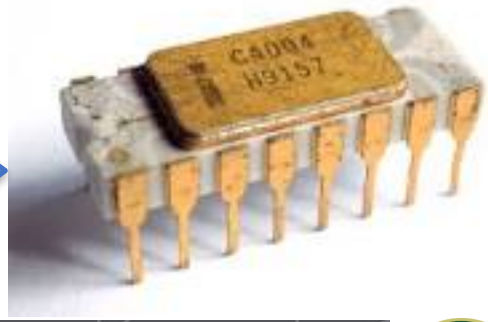
Seymore Cray

CDC-1604 - 1960: first commercially successful transistorized computer.
Speed? 0.1 million instructions/second



1960 - First IC on the market,
TIs' Multi-vibrator #502

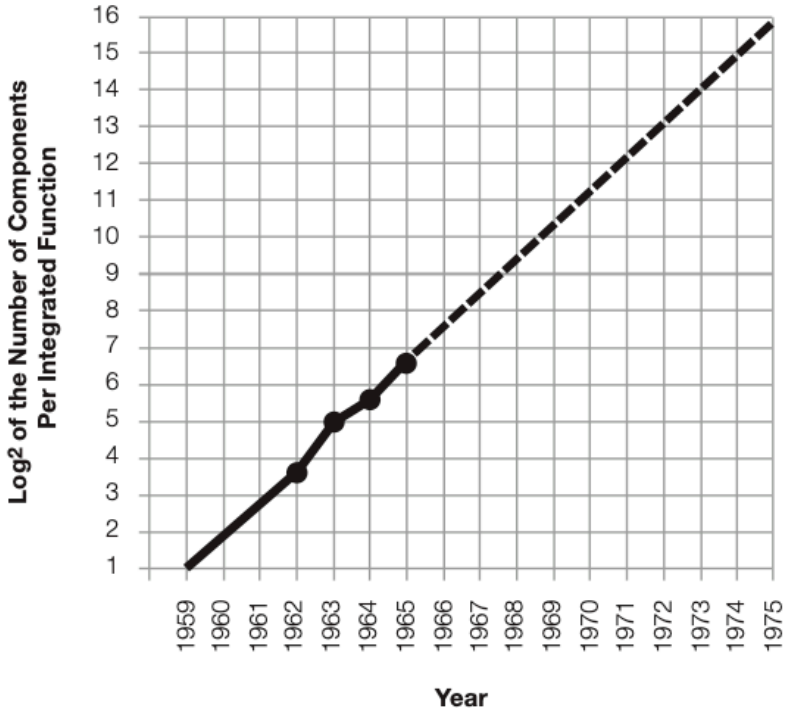
1971 - Intel 4004. First large scale IC using metal oxide silicon (MOS)



The Moore's Law Era

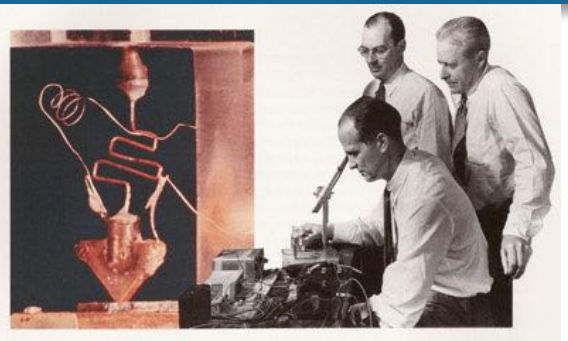
"**Moore's law** is the observation that the number of transistors in a dense integrated circuit (IC) doubles about every two years."

-Wikipedia

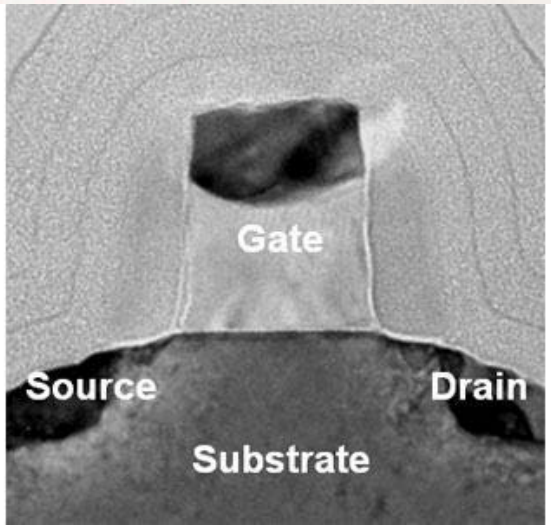


Source: Moore, G.E., "Cramming more components onto integrated circuits" Electronics, Volume 38, Number 8, April 19, 1965

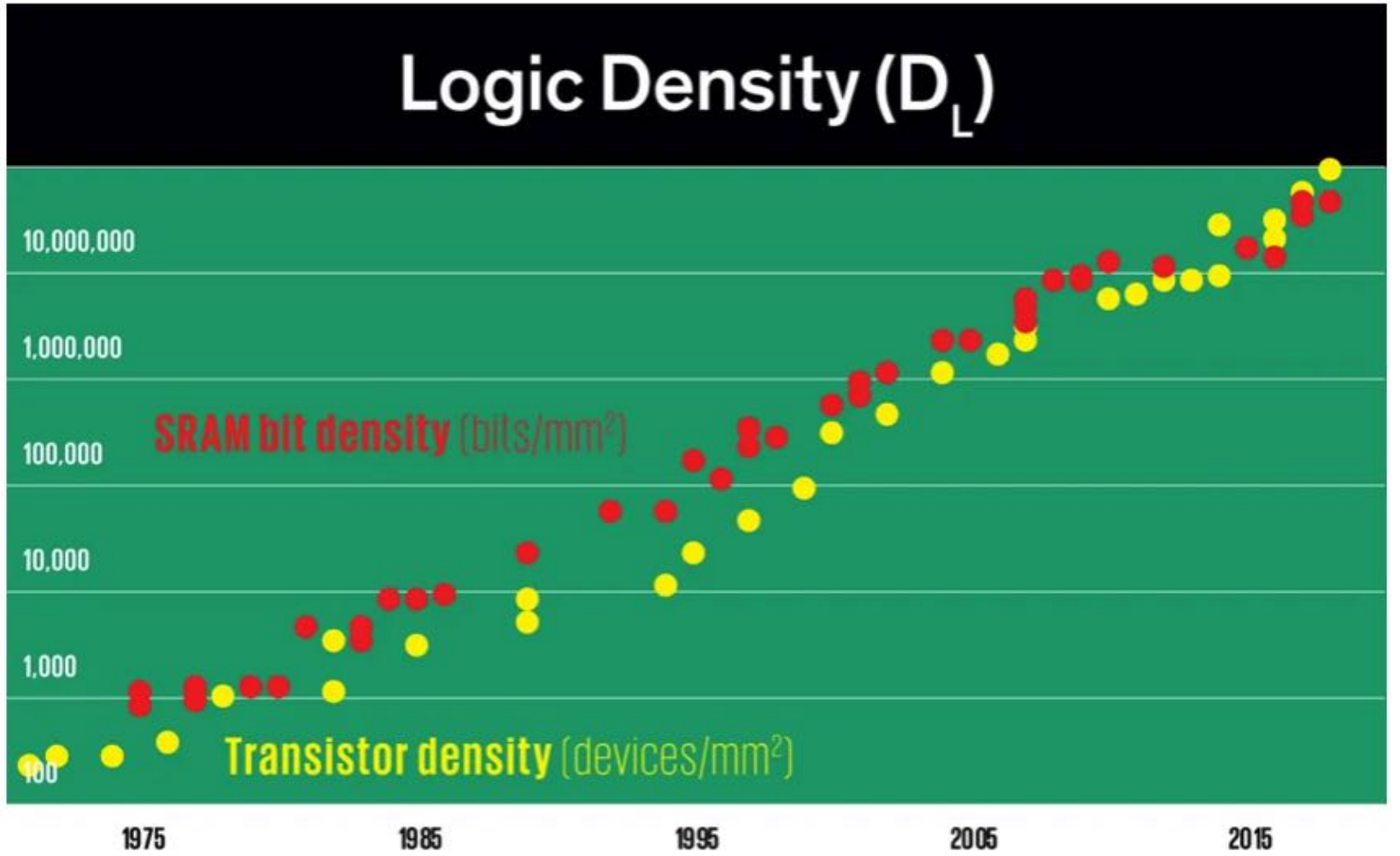
Progress in photolithography -> transistor density -> performance



1947



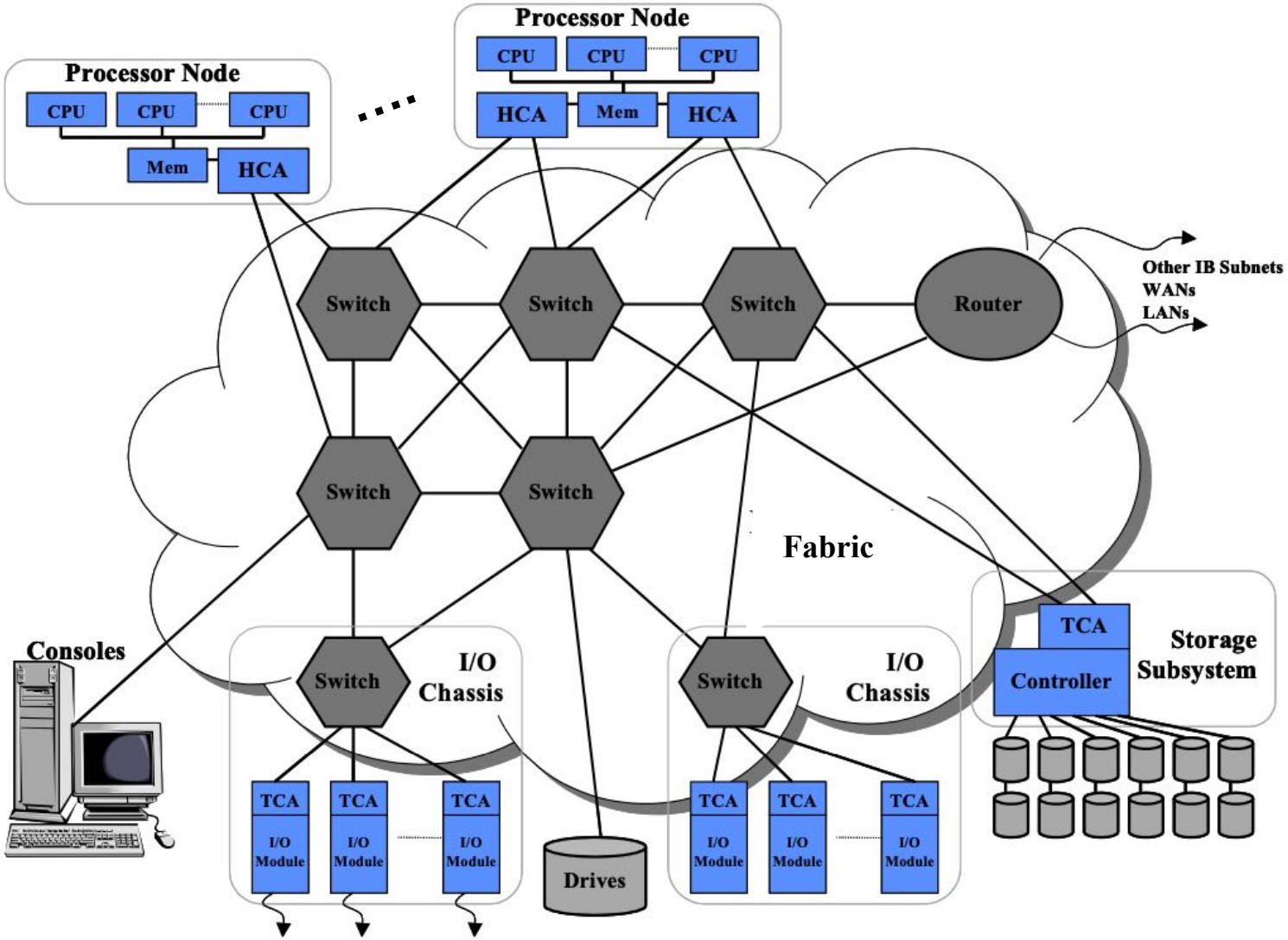
2010



Source: H. S. Philip Wong et al., " A Density Metric for Semiconductor Technology", Proceedings of the IEEE, April, 2020.

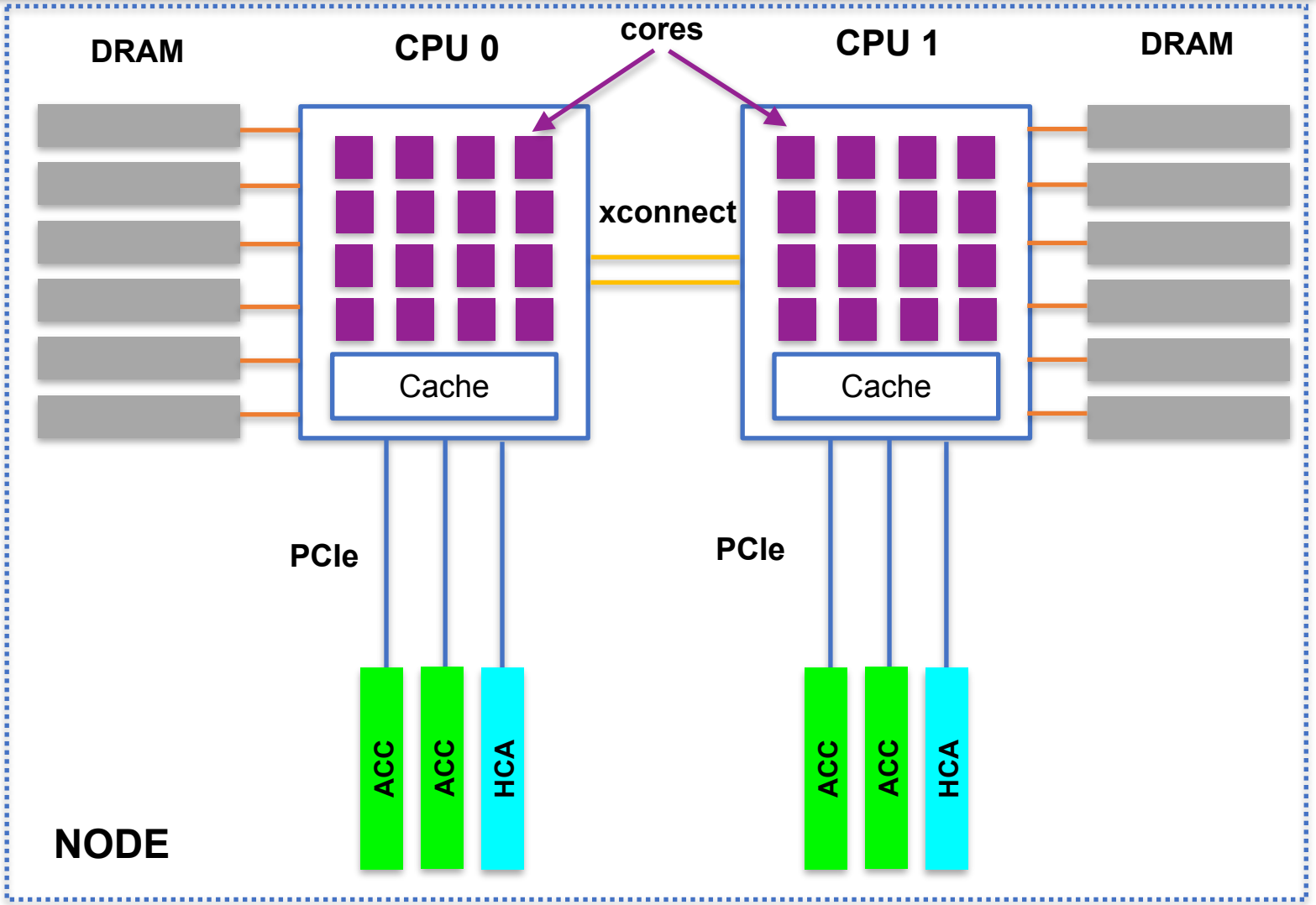
But… The wavelength of light used determines the minimum feature size

Components of a modern supercomputing cluster



Generic microprocessor node design

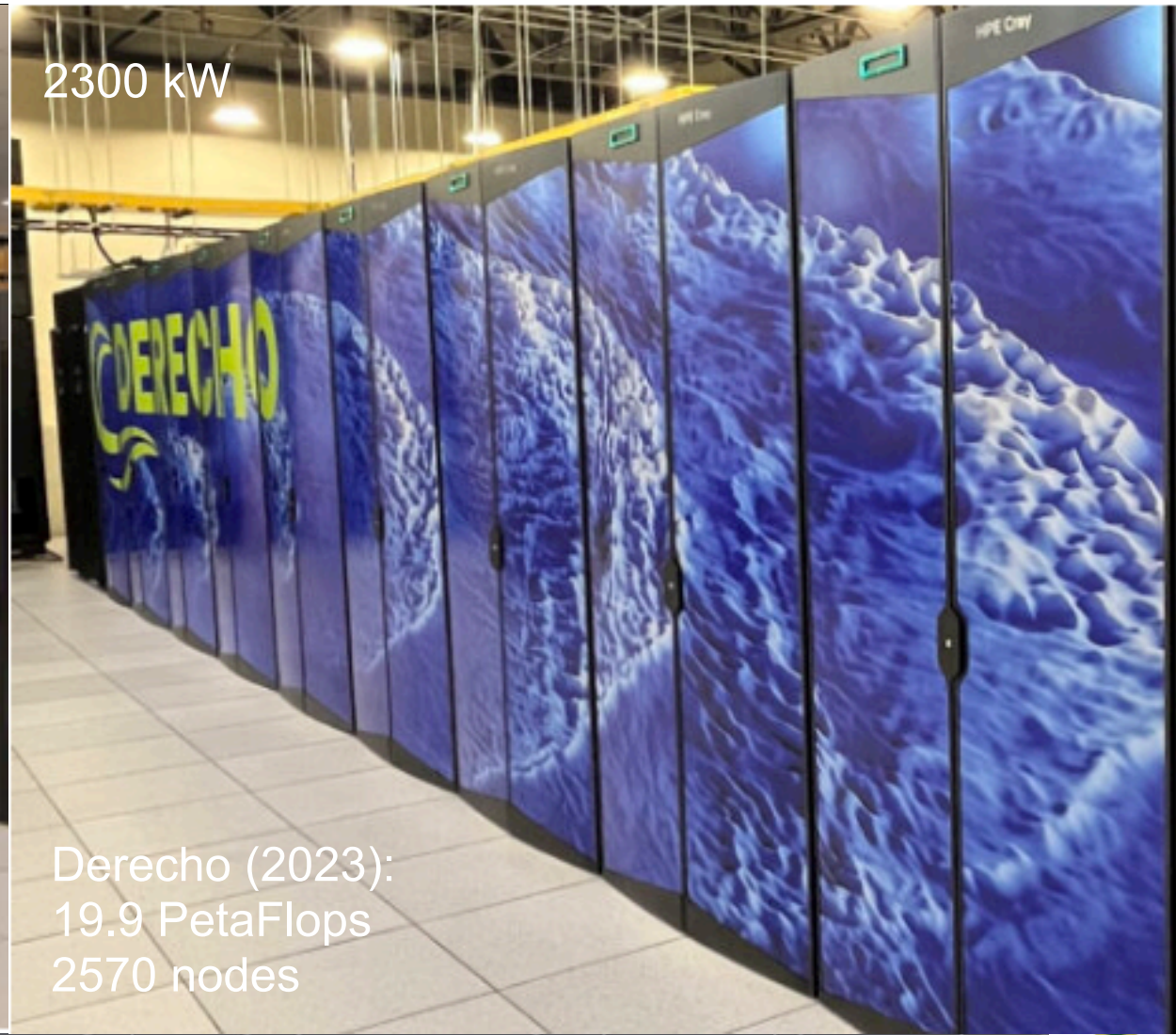
- **Microprocessor node**
  - Designed to be a general purpose host that can run an operating system itself.
- **Processing**
  - Multiple **chips** (2 shown)
  - Multiple **cores/**chip (16 shown)
  - Multiple flops per core (e.g. 16)
- **Memory System**
  - **Caches** to hide memory latency
  - **DRAM**-based memory - large/slow
  - A single shared memory space
- **Connect to other high speed devices**
  - **PCIe** slots for connectors accelerators and high speed interconnects.



HPC Building Blocks: GP-GPUs (or just GPUs)

- **General Purpose GP-GPU design features**
  - **Coprocessor accelerator**
    - Designed to be attached to a conventional microprocessor host.
    - Can’t run an operating system itself.
  - **More focus on parallelism**
    - More cores, slower clock speed.
  - **Memory size is smaller but memory bandwidth is higher.**
    - Achieved using 3D stacked high bandwidth memory (HBM)

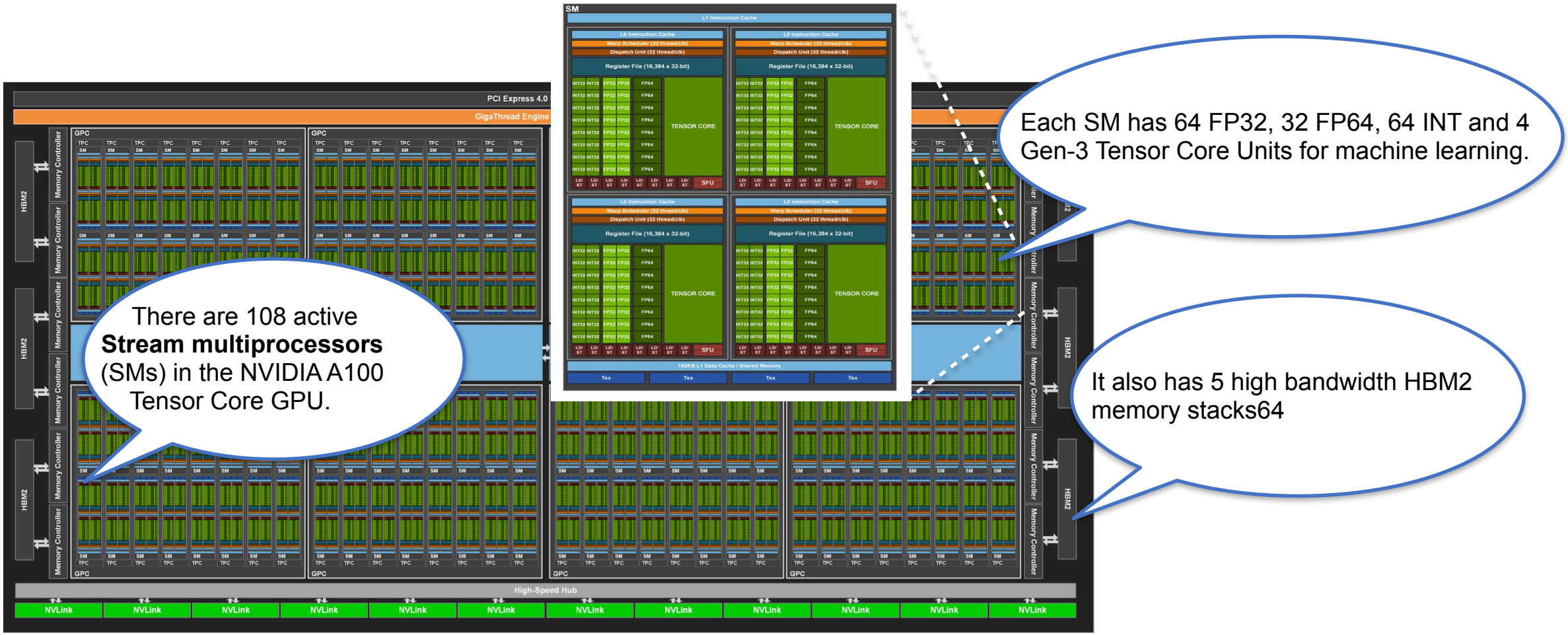
The ARC of NCAR's supercomputer at NWSC



The NCAR-Wyoming Supercomputing Center

The Derecho Supercomputer under the Skin

HPC Building Blocks: Accelerators (a.k.a. GPUs)



82 nodes with a total of 328 NVIDIA A100s accounts for 20% of Derecho's peak speed

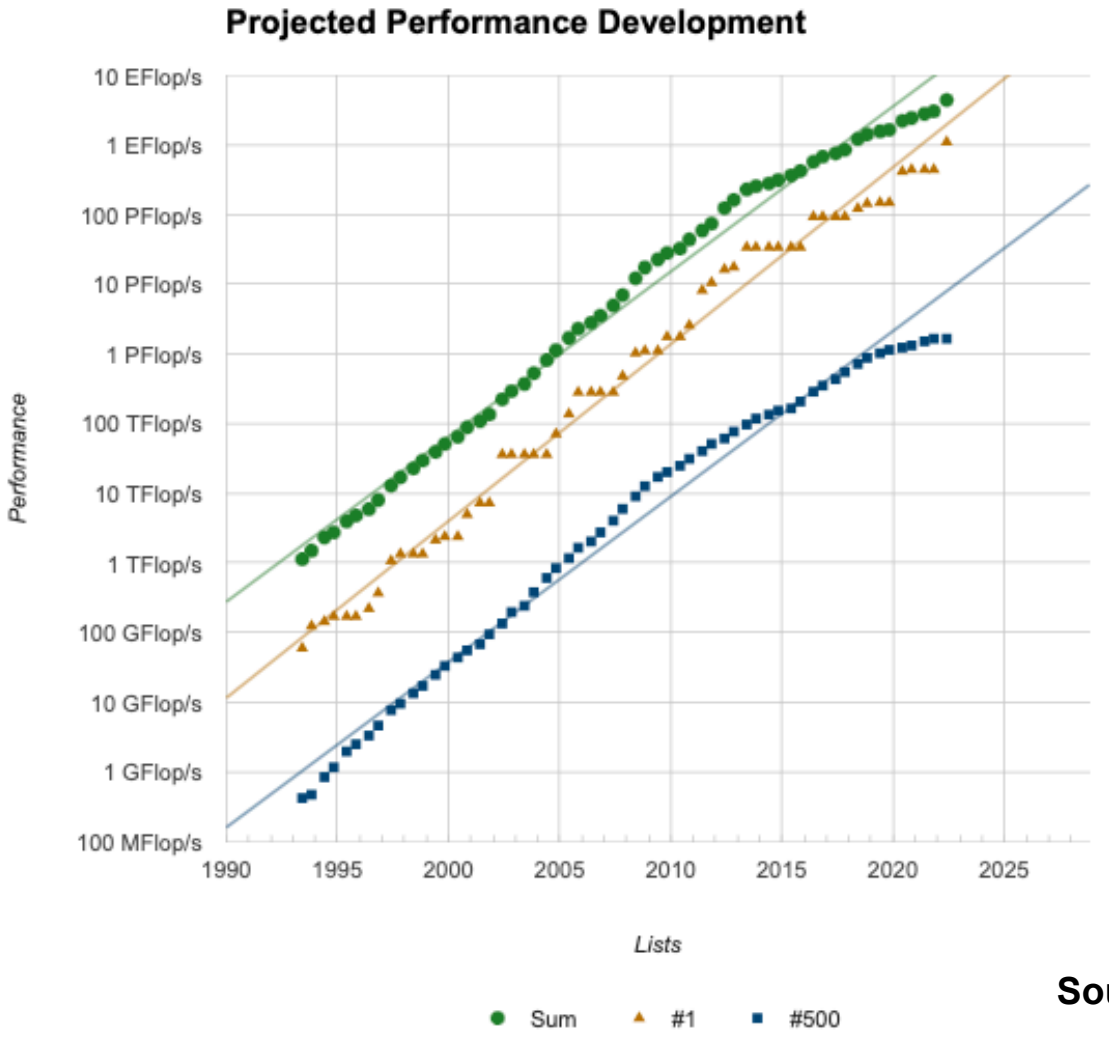
How are supercomputers ranked?

- **Recall the definition I'm using:**
  - Any of a class of computing systems that are the fastest in terms of speed of calculation available at any given time, and generally (but not exclusively) used for scientific purposes.
- **Fast, but at doing what?**
  - Traditionally, it has been the best floating point performance a system can achieve on the High Performance Linpack (HPL) benchmark (R_{max}) which solves a dense linear system (of unspecified size) in double precision floating point arithmetic (FP64).
  - This is not the same as "theoretical peak", a.k.a. R_{peak} …some machines simply cannot achieve this in practice.
  - R_{max} is used to rank systems on the Top500 list (top500.org).

Criticisms of the HPL metric

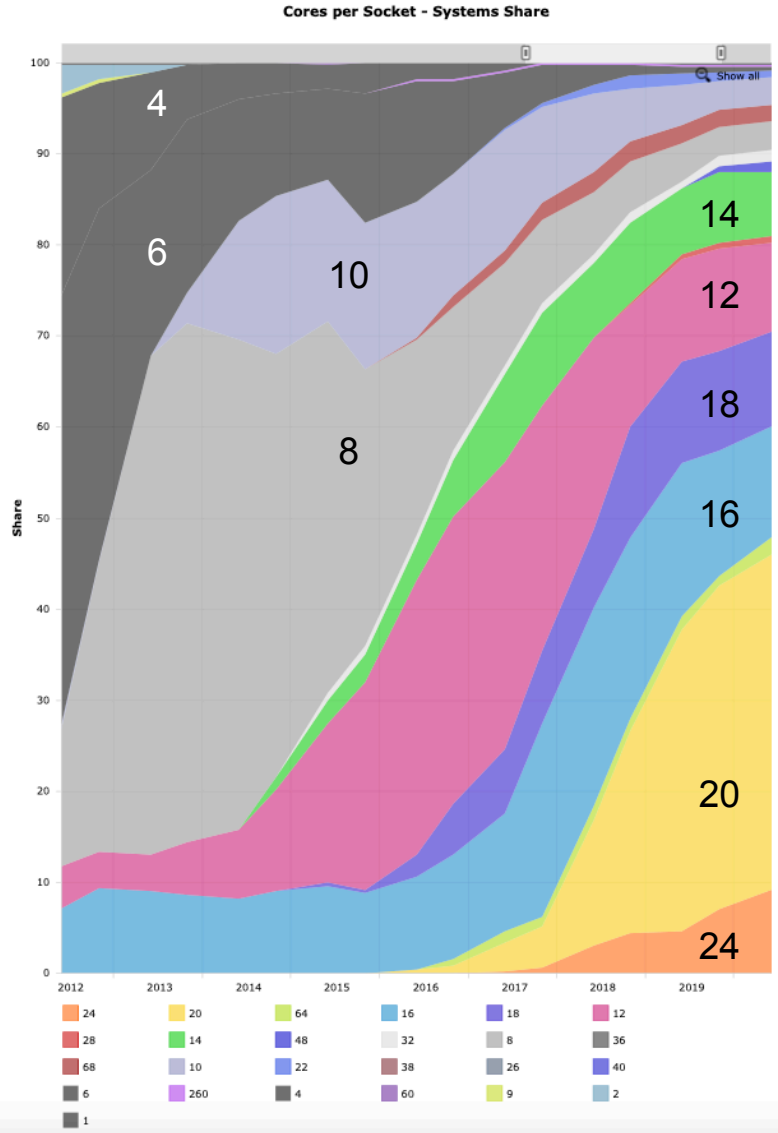
- **The HPL approach has been criticized for several reasons:**
  - **It is "flops-centric"**: many workloads are memory intensive -> **HPCG (High Performance Conjugate Gradient) benchmark.**
  - **It focuses solely on speed:** given climate change and energy costs, power efficiency (Gflops/watt) should also be a focus -> **Green500 list**.
  - **It is fixated on double precision (FP64):** many workloads (Machine Learning) are focused on reduced precision (e.g. 16-bit) -> **MLPerf Benchmark Suite**

Still, the Top500 list shows us HPC trends over time….



More cores per chip = more performance

More transistors per chip = more cores



Source: June 2022 Top 500 list

So, is the cloud a supercomputer?

- **Recall the definition I'm using:**
  - Any of a class of computing systems that are the fastest in terms of speed of calculation available at any given time, and generally (but not exclusively) used for scientific purposes.
  - So it depends on the workload.
- **Could the "cloud" replace supercomputers?**
  - A supercomputer is designed for solving tightly coupled problems, so the computing, storage and networking is typically housed at a single location.
  - With cloud computing, the computing infrastructure is distributed, and is designed to scale up loosely coupled workloads across multiple locations over the internet.
  - That said, hybrid solutions exist that try to mix the best features of each approach.

New Paradigms: Computational and Data Science

What is Computational Science?

- Computational science - an interdisciplinary field that encompasses mathematics, computer science, domain science, and software engineering to solve scientific problems on computers.
- It is also called “scientific computing” or just “modeling”.
- It our context, it is the science (art!) of using supercomputers to solve scientific problems requiring them.
- The focus of computational science historically has been deductive in nature, working out the consequences of physical law in settings not amenable to theory or experiment.
- It has been called the third scientific paradigm (after theory and experiment).

What is Data Science?

- Data science - an interdisciplinary field that combines mathematics and statistics, domain expertise, and programming skills, to extract insights from data.
- Sounds very similar to computational science!
  - Yes, but data science is inductive in nature, reasoning from specific data to infer more general conclusions. It is also called “data analytics” or “data-driven” science.
- It our context it is the science (art!) of using big data to solve scientific problems.
- It has been called the fourth scientific paradigm (after theory, experiment, and computational science).

The Five Pillars of Computational & Data Science

- Mathematics (numerical methods, statistics, physical laws)
- Computer science (algorithms, computability, parallelism)
- Architecture (processors, memory, interconnects, and storage)
- Technology (transistors, qbits, magnetic media, optical fabric)
- Software (languages, libraries, tools, and applications)

Parallel Computing

Parallel Computing: Why?

- **It would be nice to have a single processor that ran super fast to solve our problems. For one, it would be easier to program!**
- **Why not?**
  - Transistor-based digital computers have clocks, that orchestrate the computations performed by the logic gates on a chip.
  - Every time a transistor flips state, it dissipates some energy.
  - The higher the clock rate, the more heat dissipated. Thus, there is a limit to the clocking frequency beyond which the device overheats.
  - At room temperature, somewhere around 5.5 GHz seems to be the current speed limit, although 2-3 GHz is more typical in the processors we're talking about.
  - Thus, to go beyond the processing power of a single execution unit, or core, we need to run many cores in parallel to work cooperatively on different parts of the problem.
- **We need to find parallelism in our problem.**

Parallel Computing: Amdahl's Law

- **AmDahl's Law and Speed up.** Suppose you have a workload in which some fraction s of the computations must be done sequentially, and the rest (1-s) can be perfectly parallelized. Using N processing elements, the overall speedup S is given by:

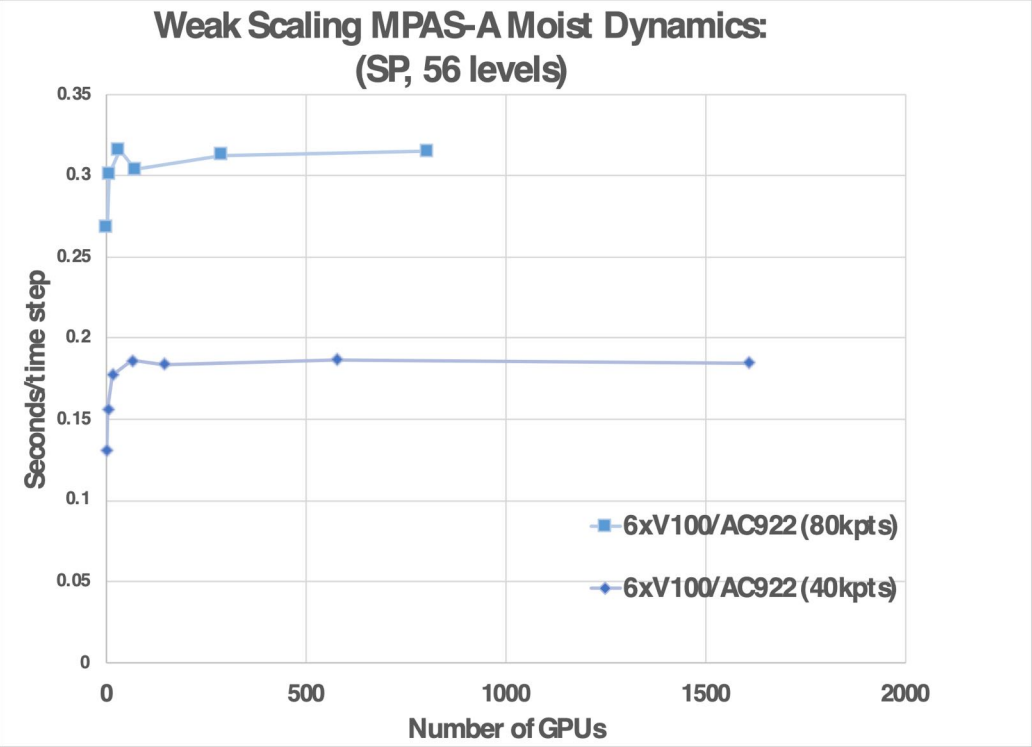
$$S(N) = \frac{1}{(s + \frac{(1-s)}{N})}; \quad S(\infty) = \frac{1}{s}$$

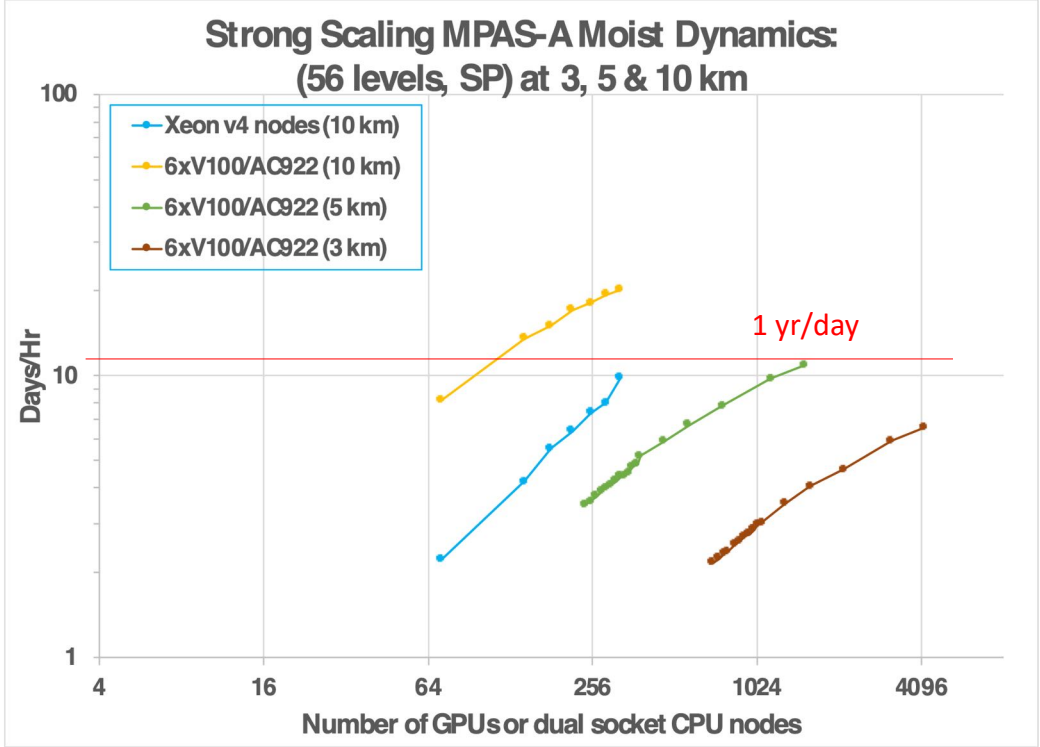
- So if your program is 1% serial (s =. 01) you can never achieve more than 100x speedup, no matter how many processing elements you throw at it.
- **Two Ingredients for Scaling (speed up)**
  - A scalable computer (no hardware bottlenecks, low communication latencies between PEs), and
  - A scalable algorithm (few communications between PEs, no serial dependencies).
    - Running completely independent work on each PE is called embarrassingly parallel.

Weak vs Strong Scaling

- **Amdahl's Law**
  - holds the problem size fixed and increases parallelism (number of PEs)
  - This is also called **strong scaling**
- **Gustafson's Law**
  - Says that increasing the problem size as the level of parallelism grows can retain scalability.
  - **This is also called weak scaling**
- If we increase the total problem size such that problem size per PE is constant, and the computational rate (flops) per PE also stays constant, the problem and the machine are said to be scalable, or to scale out.

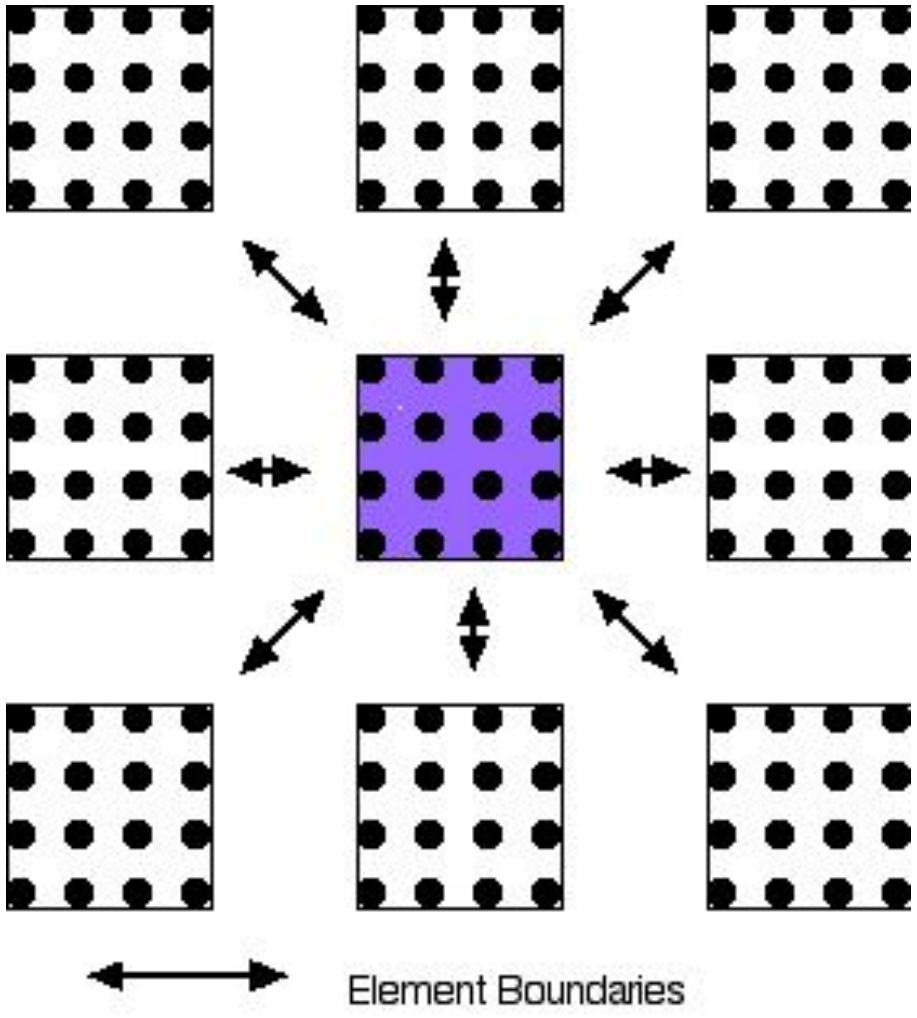
Example: MPAS-A moist dycore scaling on *Summit*[1] and *Cheyenne*[2]





[1]Benchmarking on Summit supported by DoE via an OLCF Director's Discretionary Allocation
[2]Cheyenne is a 5.4 PF, 4032-node HPE system with EDR interconnect operated by NCAR

Example: A scalable dynamical core algorithm

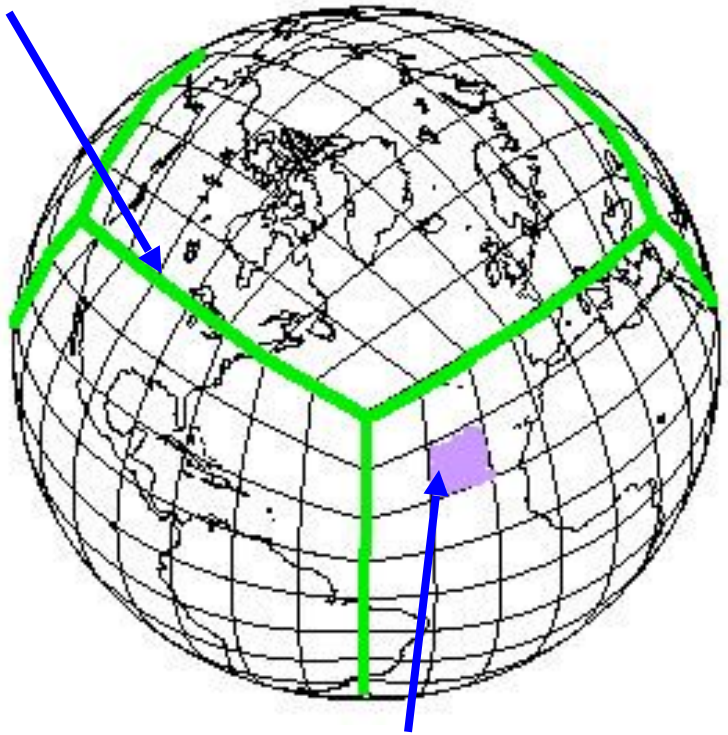


The Spectral Element Method

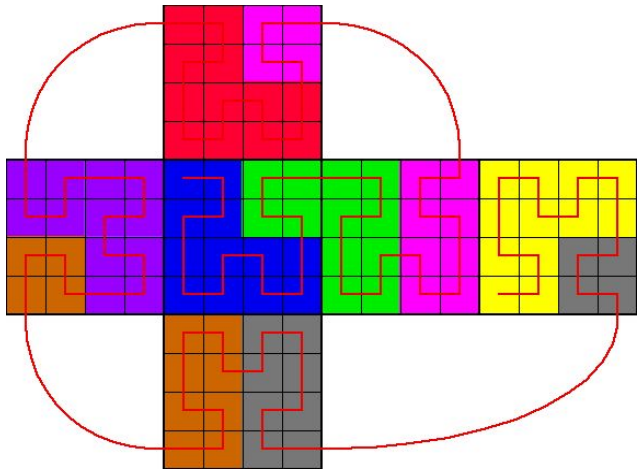
- High accuracy way to calculate derivatives (exponential convergence)
- Computationally intense local calculations on a “patch” of grid points.
- Shares only edge values of the element with nearest neighbors.

Example: Mapping Spectral Elements to the Cube Sphere

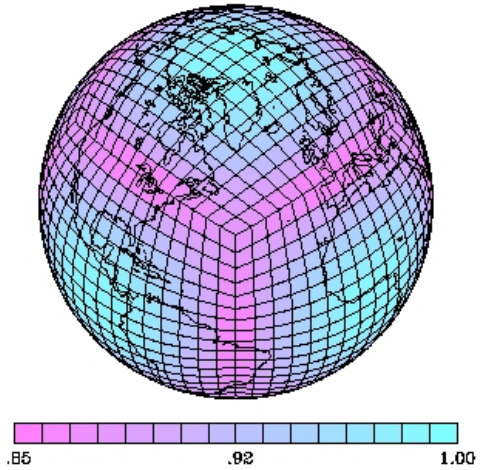
Con: Special points along edges create extra work for certain PE's (this is called a load imbalance).



Con: Metric terms to conform flat elements onto the sphere add some load imbalance to the computations.



Pro: Partitioning work using space filling curves can distribute work evenly across PEs and minimize communications.



Pro:The cube sphere grid is quasi-uniform - little bunching up of points.

10 Essentials for Getting Started

3 Essentials for Using HPC Resources

- **Learn Unix command line operations**
  - How to print working (**pwd**), make (**mkdir**), and change directories (**cd**);
  - How to list (**ls**); copy (**cp**); move (**mv**); and remove* (**rm**) files;
- **Learn the batch queueing systems (many flavors - PBS shown here)**
  - How to kill (**qdel**), submit (**qsub**), and get the status (**qstat**) of a batch job (in that order)
  - Learn how to request an interactive session.
  - Learn to write shell scripts for batch (non-interactive) sessions (many flavors)
- **Learn how to use environment modules**
  - How to **list**, **load**, **unload**, **swap**, and **purge** (clean) your module environment
  - e.g. how to load *exactly* what you want: > **module load** *<tool_name>/i.j.k*

*Never type rm after midnight.

4 Essentials for Accessing & Transferring Data to HPC Resources

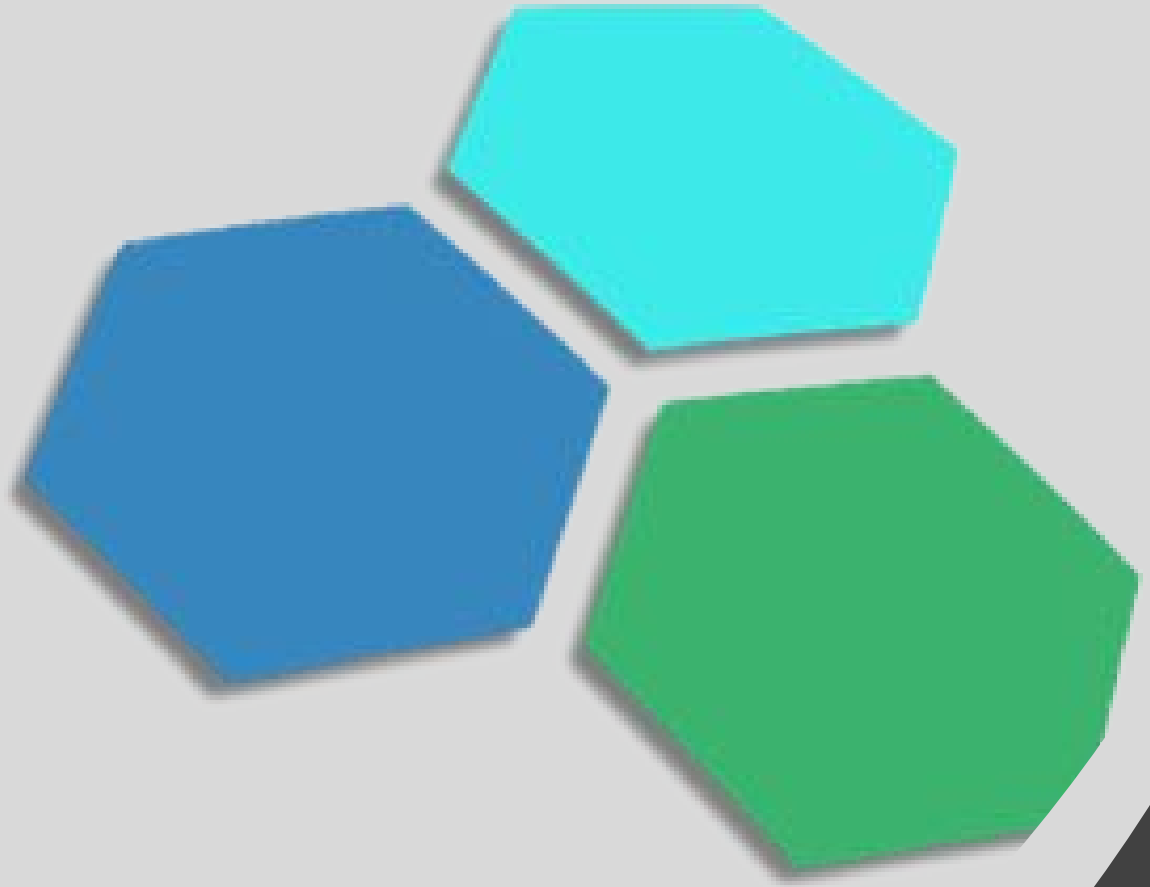
- **Learn ssh and TOTP (Time-based One Time Password) for accessing the system**
  - **ssh -X** *<username>***@cheyenne.ucar.edu**
  - Learn how to use TOTP flavors (apps on your phone): **Duo mobile, Google Authenticator, MobilePASS+**
  - How to list (ls); copy (cp); move (mv); and remove (rm) files;
- **Learn how to use scp command (small scale transfers)**
  - To: >**scp** <lclpath>/*.dat <remuser>**@remsystem.univ.edu:**<rempath>
  - From: >**scp** <remuser>**@remsystem.univ.edu:**<rempath>/*.dat <lclpath>
- **Learn how to use Globus to transfer files between computing centers**
  - Create a Globus ID (globusid.org)
  - Install **Globus Connect Personal** on your laptop.
  - Setup your own Globus endpoint, find the endpoints of the systems you work with.
- **Learn other transfer tools as necessary:**
  - PuTTY utilities (Windows systems)
  - FileZilla (multiplatform sftp solution)

3 Essentials for Data Analysis and Visualization

- **Learn the Python Ecosystem**
  - How to program the Python language itself
  - **Numpy** -math functions, arrays - faster than regular python arrays
  - **SciPy -** numerical algorithms (Linear algebra, Stats)
  - **matplotlib** (Visualization)
  - **scikit-learn** (Machine Learning)
- **Learn the Conda package management system**
  - How to **create**, **activate**, **install**, **deactivate**, **remove** Conda environments
- **Learn Jupyter Notebooks to document & share your work**
  - Contains both executable (e.g. python) and
  - human readable documents (text, equations, plots).

The Eleventh Essential: How to get an allocation

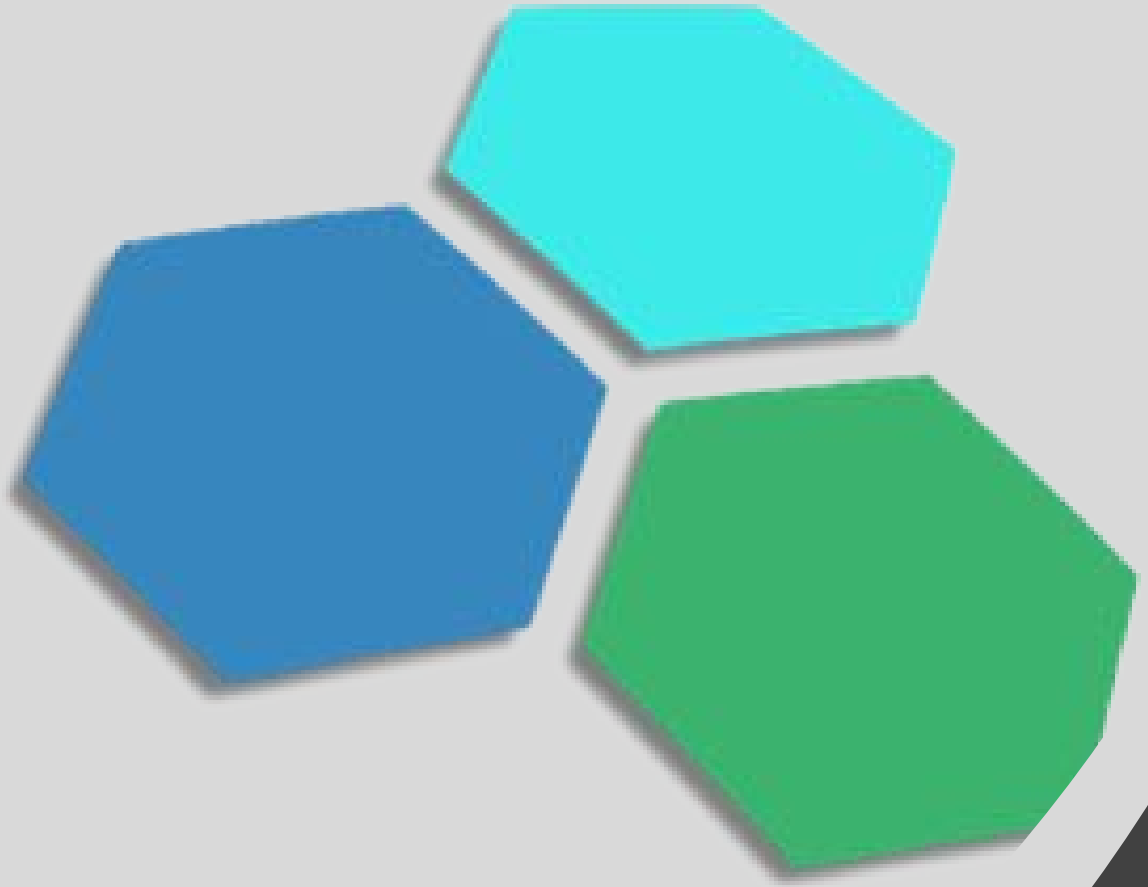
- **Get added to your advisor's account**
  - Solves the allocation request issue.
  - OJT is OK, but…
- **Take HPC training classes/courses**
  - Many centers offer classes with hands-on usage of systems with training acct.
  - Network!
- **Most centers offer "startup" or "exploratory" allocations**
  - You have a research idea… but no grant.
  - Typically short term and small in size, but have a low bar.
  - You still may need some form of faculty sponsorship
- **Learn how to write a good resource allocation request**
  - **Intellectual Merit:** Explain the research question and why it needs a supercomputer to answer it. Don't confuse an HPC resource request with a science proposal.
  - **Methodology:** Make sure the methodology (i.e. models, tools) is appropriate for this HPC resource.
  - **Research Plan:** Do the computational experiments answer the question, Are your resource estimates detailed and credible?
  - **Efficient Use:** Will the resources being used efficiently? Could the runs begin from a pre-initialized state?



Questions

Contact Information
richard.d.loft@areanddee.com

Backup Slides



Backup Slides

Contact Information
richard.d.loft@areanddee.com

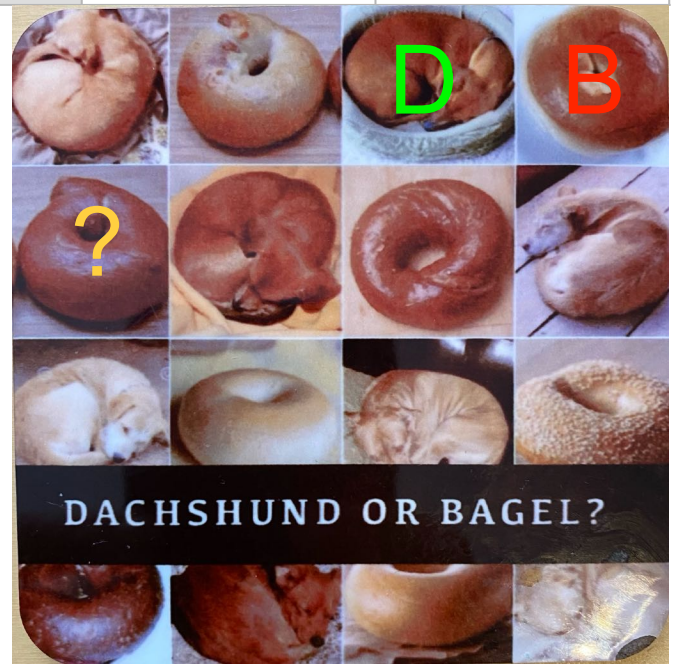
Machine Learning

Machine Learning

AI is the capability of a machine to imitate intelligent human behavior. Here I focus on a subset of AI - learning algorithms.

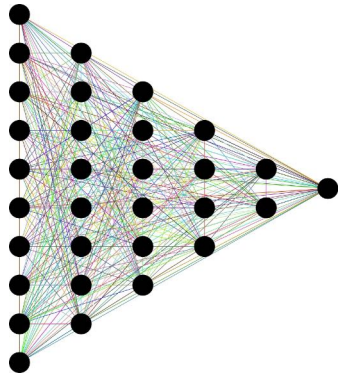
- **Machine Learning** Algorithms that improve their performance at some task with experience.
  - The experience is provided by **training data.**
- **Supervised learning** means the training data is labelled.
  - **Unsupervised learning** means that it is not.
- **Discrete means**
  - Is this is a Dachshund or Bagel?
  - There is no continuum here… it is either one or another.

	Supervised	**Unsupervised**
Discrete	Classification	Clustering
Continuous	Regression	Dimensionality Reduction



Machine Learning Algorithmic tour…

- **Clustering algorithms** (e.g. K-means)
- **Classifiers**
  - Neural networks
  - Random Forests
  - Baysian networks
  - Support vector machines
  - K nearest neighbor

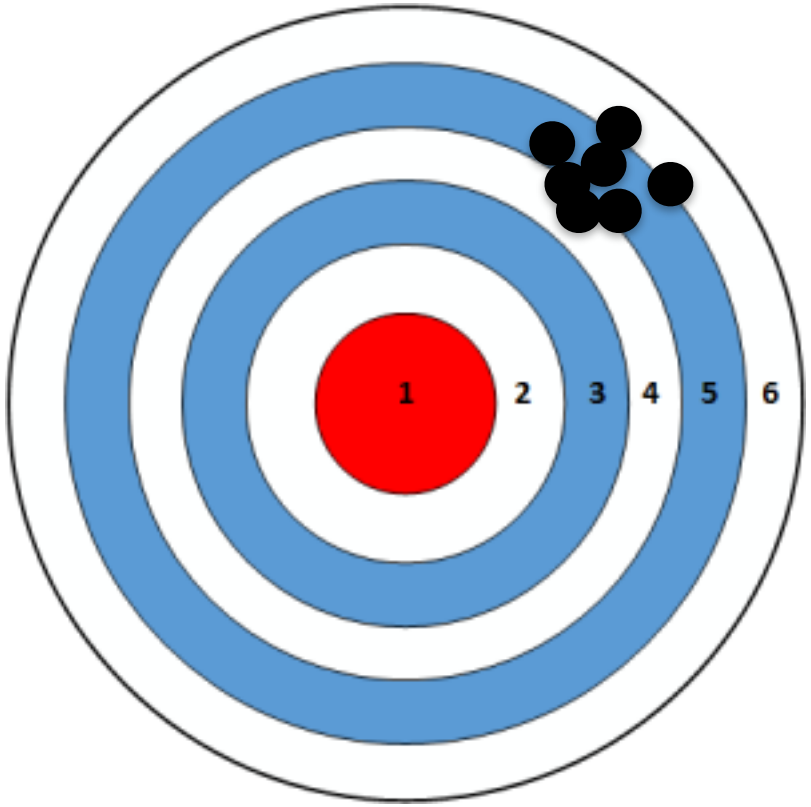


Dachshund
Bagel

- **Dimensionality Reduction**
  - **The curse of dimensionality:** The space of possible parameter values grows factorially as the number of features in the data set grows.
  - Only keep most important features (backwards elimination)
  - Create combinations of new features (e.g. principal component analysis)

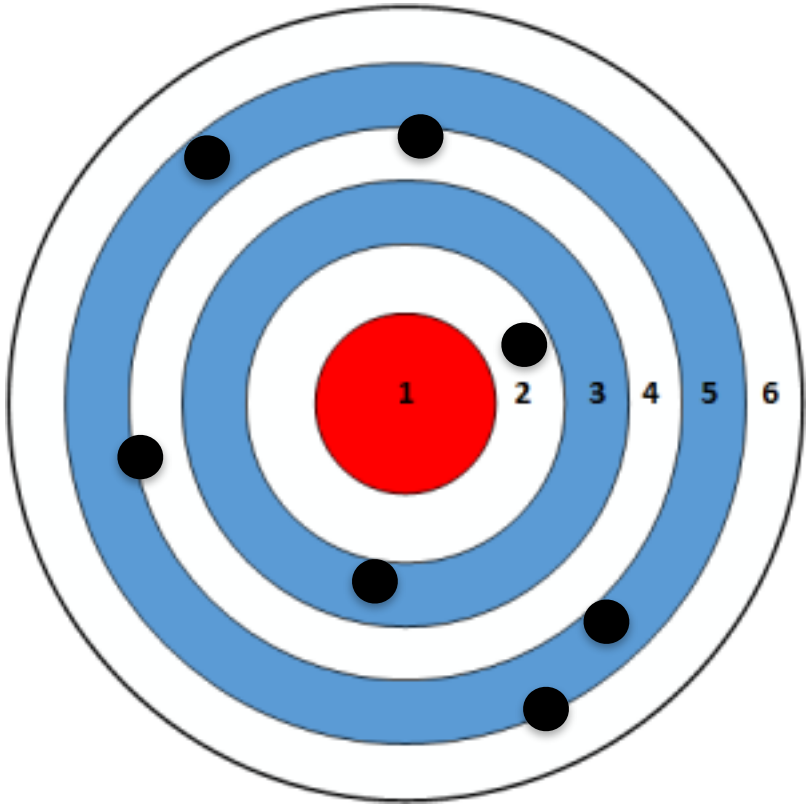
	Supervised	**Unsupervised**
Discrete	Classification	Clustering
Continuous	Regression	Dimensionality Reduction

Machine Learning: Over- and under-fitting



Models with too few parameters are inaccurate because of a large bias and a small variance. They are under-fitting.

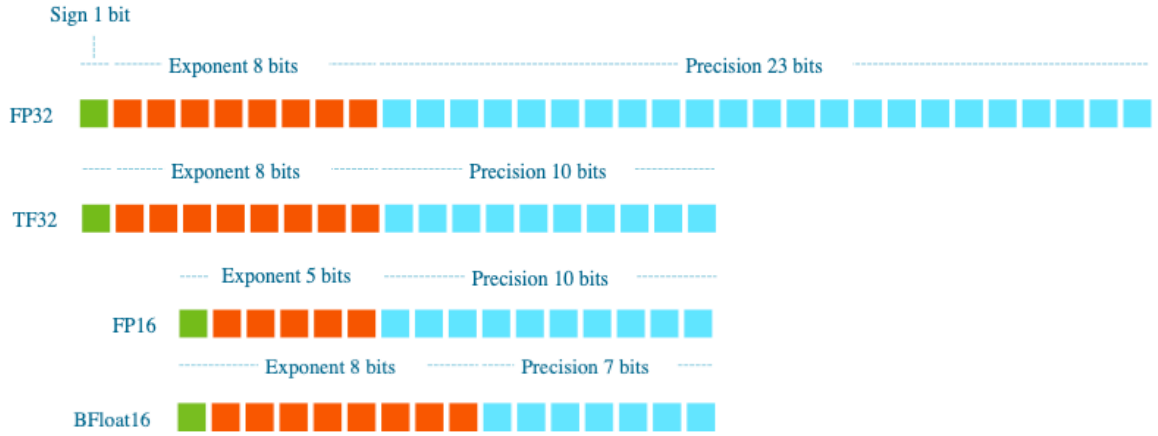
ML Models with high variance and low bias have too many parameters. They are over-fitting.



Machine Learning: Hardware/Software Ecosystem

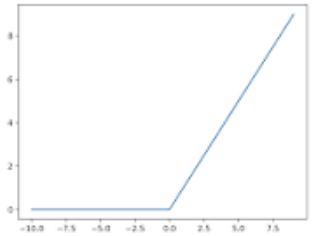
- **Special Purpose Hardware**
  - **NVIDIA GPUs - Tensor Cores** Tensor cores are processing units that accelerate the process of matrix multiplication.
- **Precision**
  - ML architectures are moving towards lower precision
  - **Training -** higher precision required **(FP32 -> BF16)**
  - **Inference -** lower precision required **(BF16 -> INT8)**
- **Software -** unlike traditional HPC, most ML work is framework based, rather than writing bespoke models.
  - **ML Frameworks**
    - scikit-learn
    - Tensorflow/Keras-most popular for deep learning
    - PyTorch
  - **Horovod -** distributed deep learning framework

Some Floating Point Numbers used in Machine Learning



Tensorflow/Keras code to build a neural network image classifier

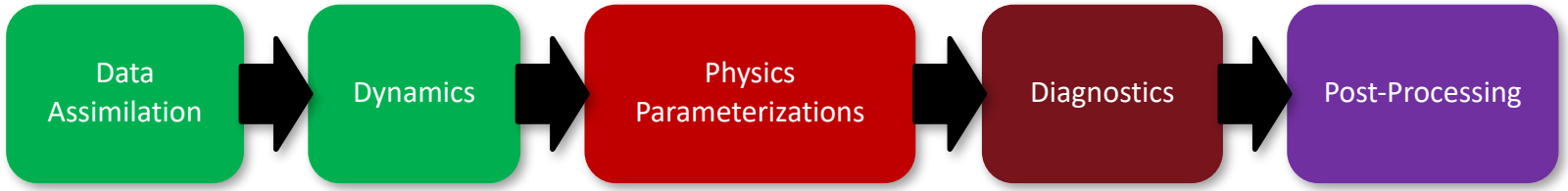
```
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10)
])
```



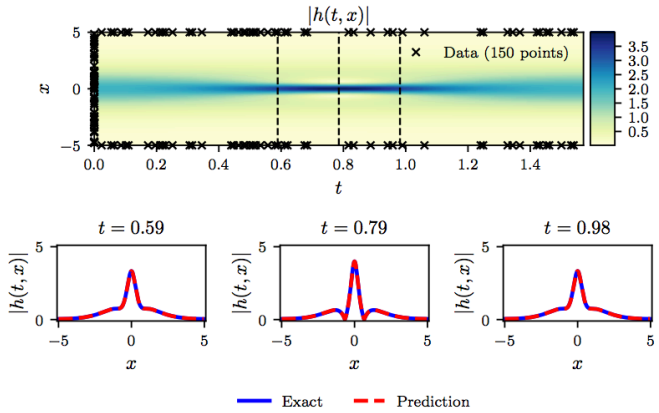
The RELU activation function

Applications of Machine Learning to Earth System Science

Machine Learning Along the Science Pipeline

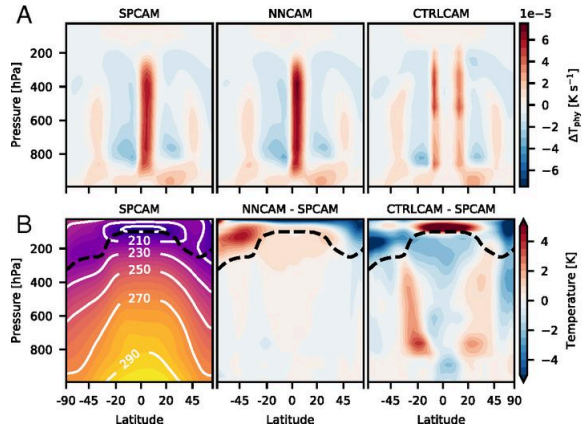


Physics-Informed Neural Network PDE Solvers



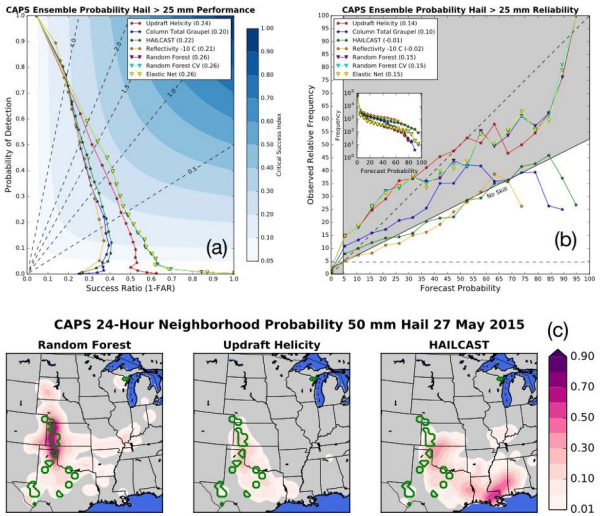
Raissi, M., P. Perdikaris, and G. E. Karniadakis, 2017

Neural Network Parameterization



S. Rasp, M. S. Pritchard, and P. Gentine, 2018

Machine Learning Hail Prediction

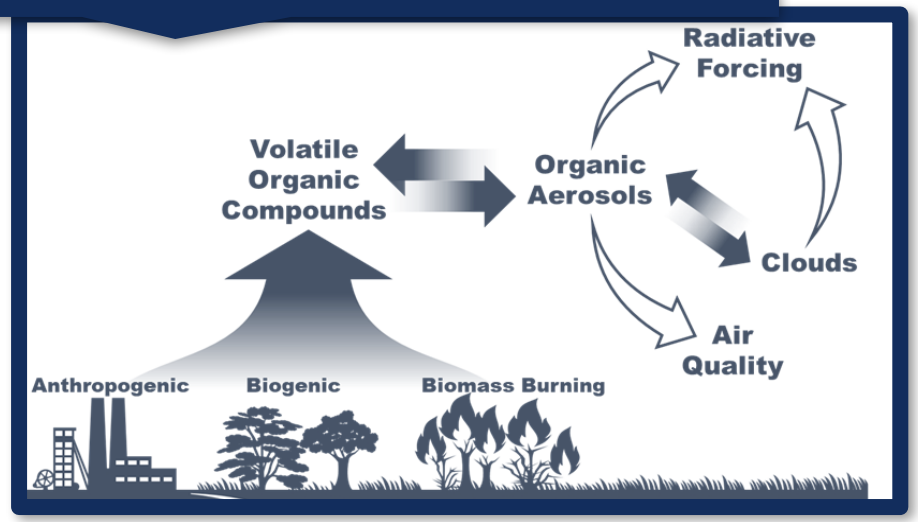


D. J. Gagne et al, 2017

Slide Credit: D.J. Gagne, NCAR

How Machine Learning is taming the complexity of organic aerosol chemistry

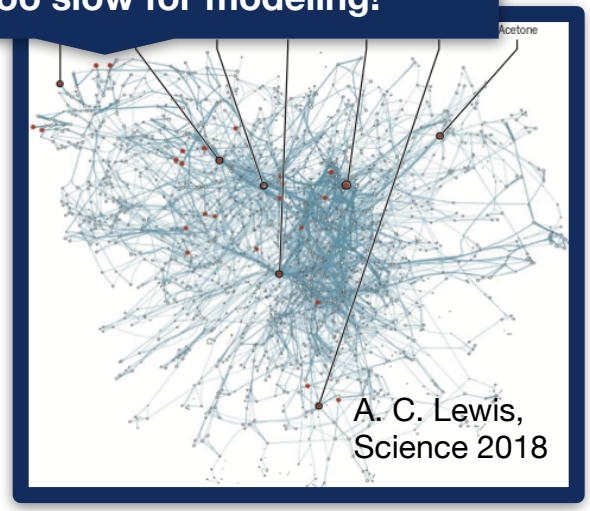
The driver: Understand urban air pollution



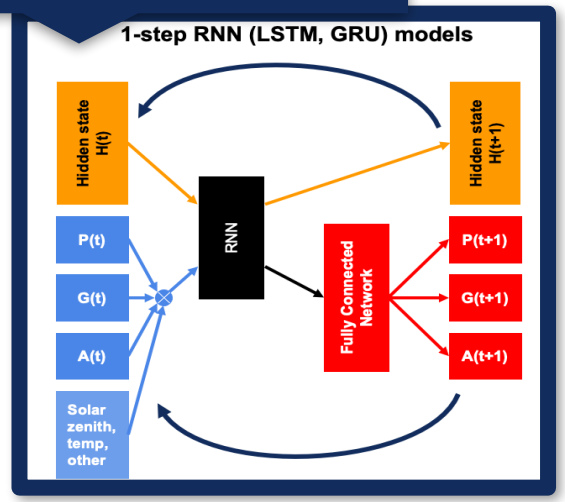
Project status:

- Testing within the GEOS-Chem model
- RNN inference is **O(10^3–10^4) faster** than GECKO-A

Complex reaction network: Too slow for modeling!



Training Recursive NN



Results: RNN vs GECKO-A (toluene sample)

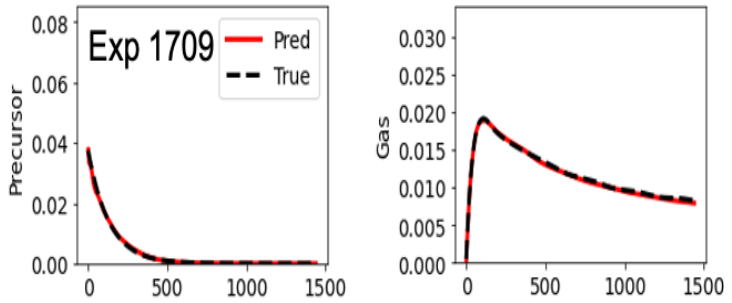


Image: courtesy of AIML group, NCAR

Replacing cloud rain processes with Neural Networks

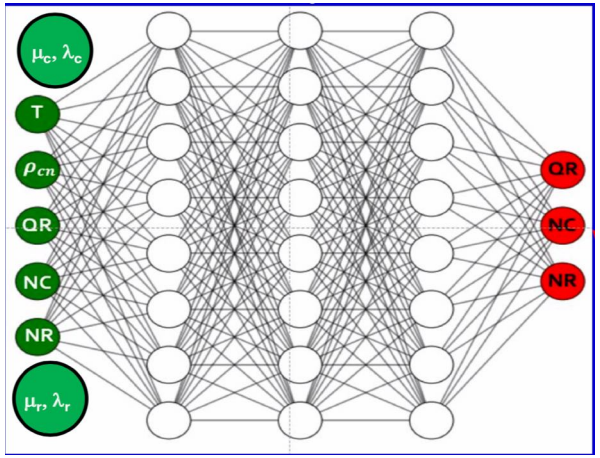
ML TAU emulator:

3 classifier networks, 4 regression networks

82,327 weights total

Dense Neural Network Hyperparameters

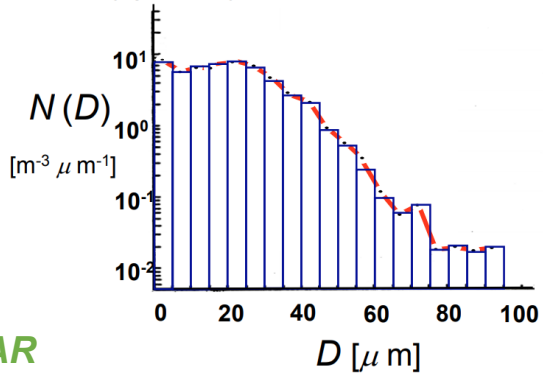
- 4 layers
- 60 neurons per hidden layer
- 11,761 total weights
- Rectified Linear Unit (ReLU) activation fns



Bin Scheme (Tel Aviv University (TAU):

- Divide particle sizes into bins **(35)**
- Calculate evolution of each bin separately
- Better representation of interactions
- Much more computationally expensive than current bulk schemes used in climate models

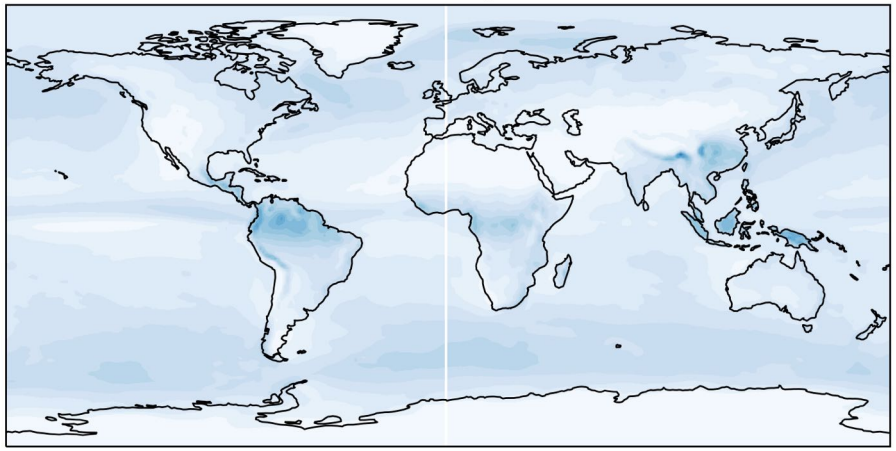
Bin-resolving: (spectral) $N(D) = \sum_{i=1}^{I} N_i$



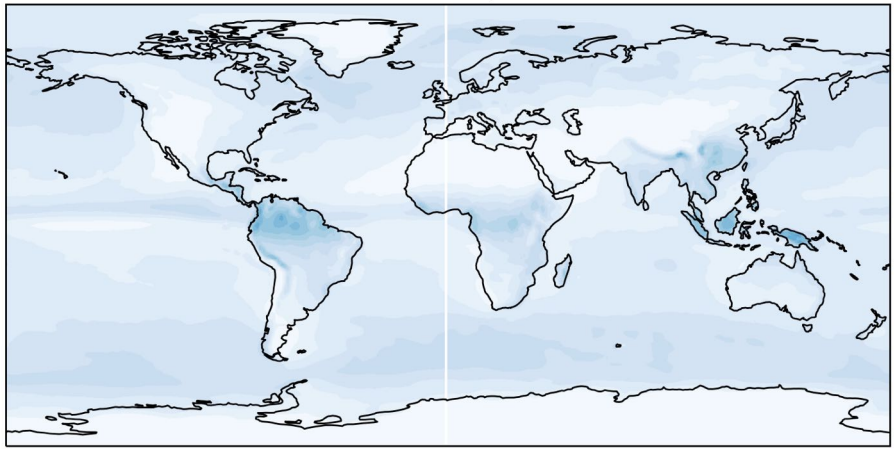
Slide Credit: DJ Gagne, NCAR

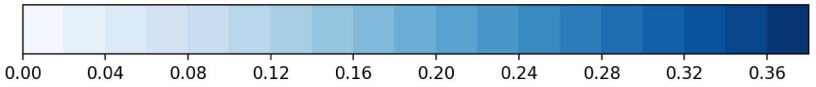
Results after 9 simulated years

ML TAU Mean Cloud Liquid Water Path



TAU Bin Mean Cloud Liquid Water Path





Future Technology

An “All in Deep Learning” Approach to NWP

Adaptive Fourier Neural Operator (AFNO) model

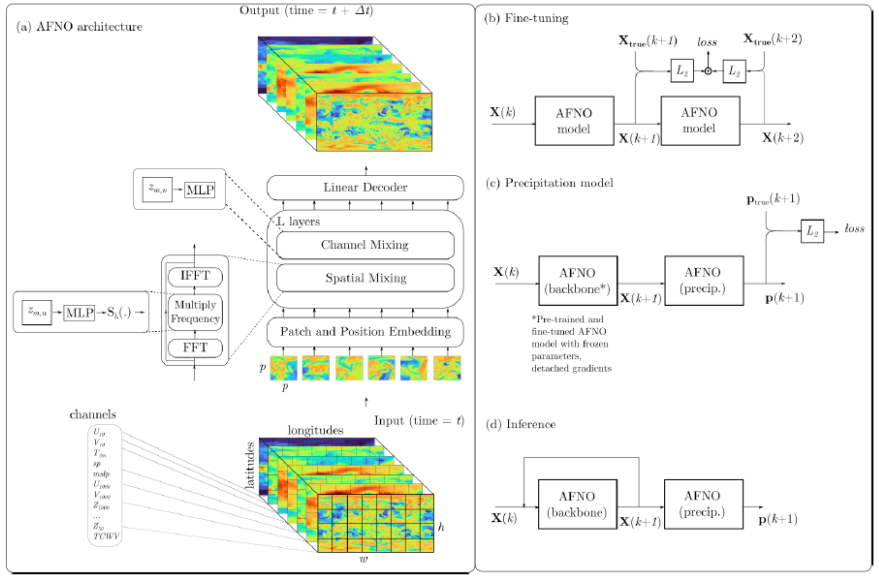


Figure 2: (a) The multi-layer transformer architecture that utilizes the Adaptive Fourier Neural Operator with shared MLP and frequency soft-thresholding for spatial token mixing. The input frame is first divided into a $h \times w$ grid of patches, where each patch has a small size $p \times p \times e$. Each patch is then embedded in a higher dimensional space with high number of latent channels and position embedding is added to form a sequence of tokens. Tokens are then mixed spatially using AFNO, and subsequently for each token the latent channels are mixed. This process is repeated for L layers, and finally a linear decoder reconstructs the patches for the next frame from the final embedding. The right-hand panels describe the FourCastNet model's additional training and inference modes: (b) two-step fine-tuning, (c) backbone model that forecasts the 20 variables in Table 1 with secondary precipitation diagnostic model (note that $\mathbf{p}(k+1)$ denotes the 6 hour accumulated total precipitation that falls between $k+1$ and $k+2$ time steps) (d) forecast model in free-running autoregressive inference mode.

Credit: Pathak, J., et al. “FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators”. ***unpublished***
arXiv:2202.11214

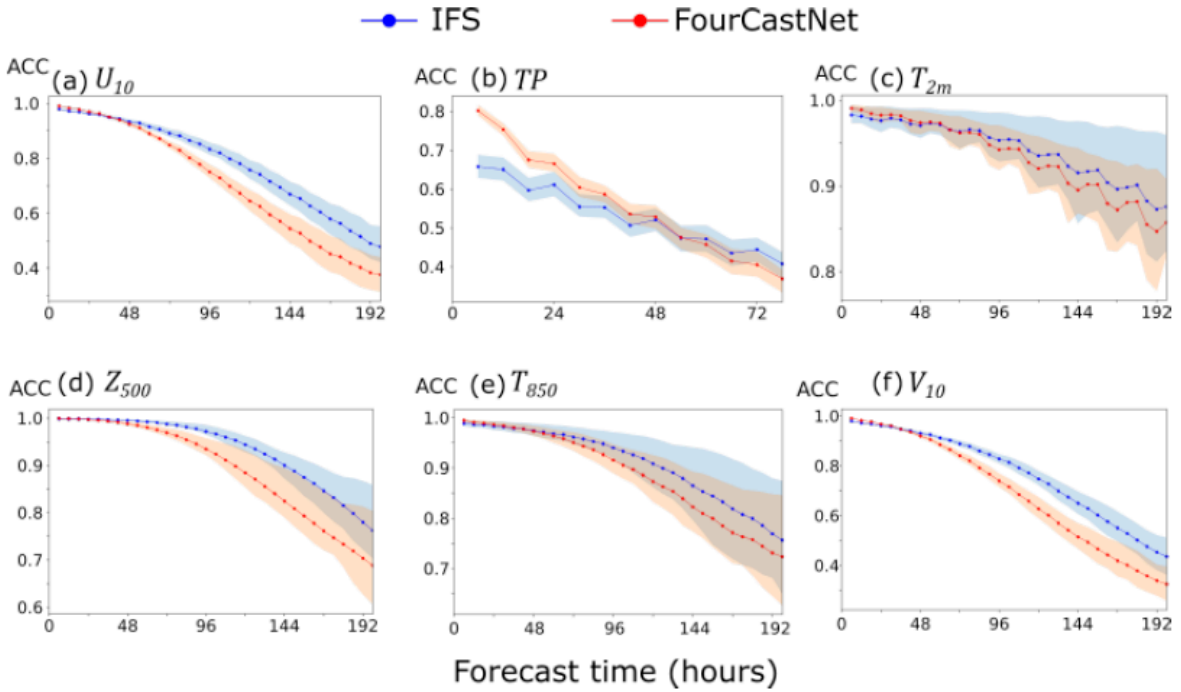
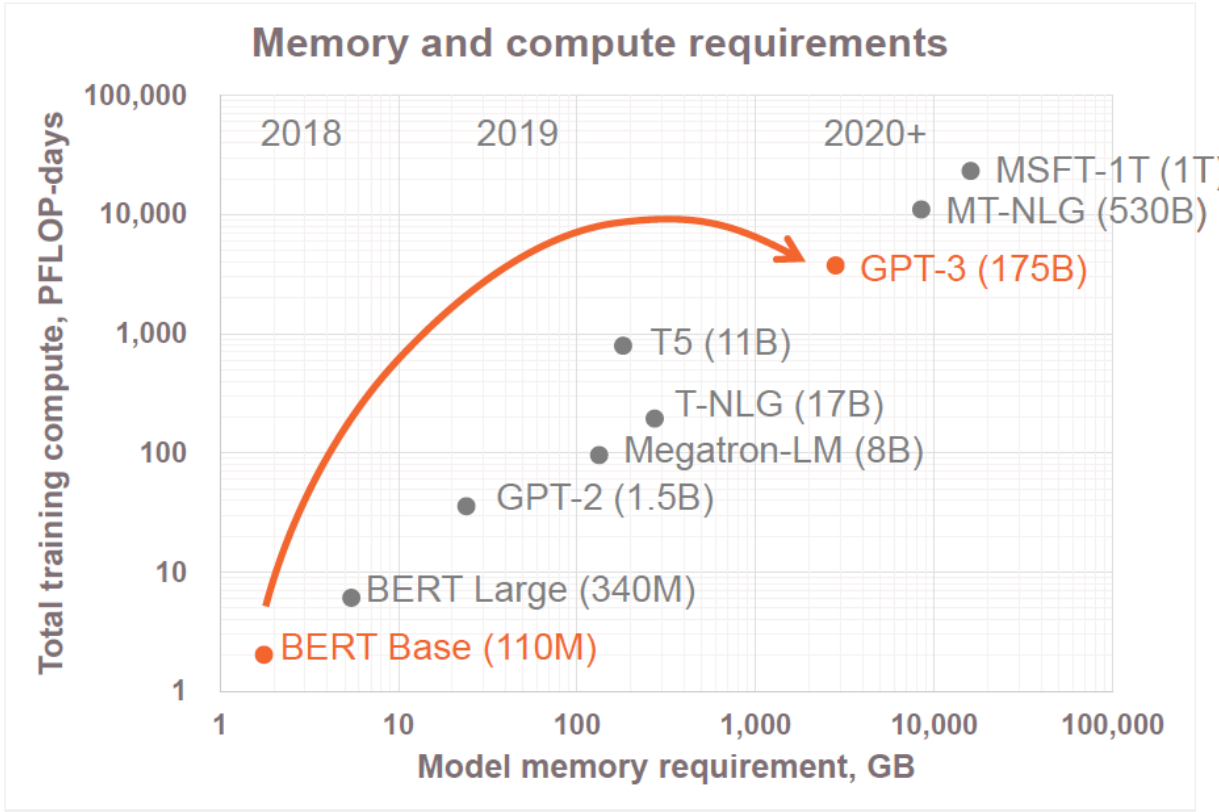


Figure 6: Latitude weighted ACC for the FourCastNet model forecasts (red line with markers) and the corresponding matched IFS forecasts (blue line with markers) averaged over several forecasts initialized using initial conditions in the out-of-sample testing dataset corresponding to the calendar year 2018 for the variables (a) U_{10}, (b) TP, (c) T_{2m}, (d) Z_{500}, (e) T_{850}, and (f) V_{10}. The ACC values are averaged over N_f initial conditions over a full year with an interval of D days between consecutive initial conditions to account for seasonal variability in forecast skill. The N_f and D values are specified in Table 4. The appropriately colored shaded regions around the ACC curves indicate the region between the first and third quartile values of the corresponding quantity at each time step. We also plot the latitude weighted RMSE curves for the FourCastNet and IFS models in Figure 13 in Appendix D.

FourCastNet throughput @ 25 km is 14 simulated days per second (SDPS)

Huge Neural Network models now driving architecture

Exponential Growth of Neural Networks



Over 1000x increase

In just **2 years**

Tomorrow, **multi-trillion** parameter models

cerebras

Slide Credit: Sean Lie, Cerebras Systems @ HotChips 34

All-in ML Wafer-scale chips

Cerebras Wafer-Scale Engine (WSE-2)

The Largest Chip in the World

850,000 cores optimized for sparse linear algebra

46,225 mm² silicon

2.6 trillion transistors

40 gigabytes of on-chip memory

20 PByte/s memory bandwidth

220 Pbit/s fabric bandwidth

7nm process technology

56x larger than largest GPU

cerebras

Slide Credit: Sean Lie, Cerebras Systems @ HotChips 34

Quantum Computing

Quantum Computing: the promise

- Conventional systems use bits: 1 or 0
- Quantum computer is based on qbits - a system which occupies two states at the same time. i.e. a spin with an orientation that is both up (1) and down (0):

$$\psi = a|0> + b|1> ; |a|^2 + |b|^2 = 1$$

$|a|^2$ expresses the probability measuring 0

$|b|^2$ expresses the probability measuring 1

- Superposition is a consequence of the wave nature of the Schrödinger Equation.
- So where's the computational beef here? Entanglement.
  - Entanglement is a quantum phenomenon in which, for example, 2 qbits act as a single 4 state system.
  - If we can prepare (entangle) a state containing many qbits (say N) that describes a system we are interested in, and operate on it, we can evaluate 2^N possible states simultaneously.
  - This is quantum parallelism, and it can be enormous.
- Not surprisingly, there is a major push to increase the number of entangled qbits, and our control over their state.

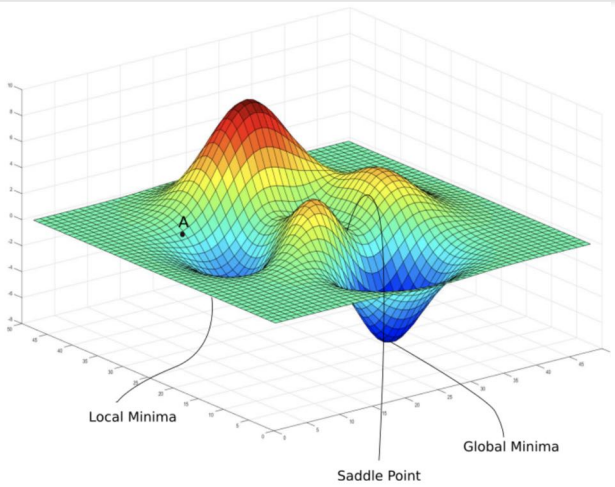
Quantum Computing: the catch(es)

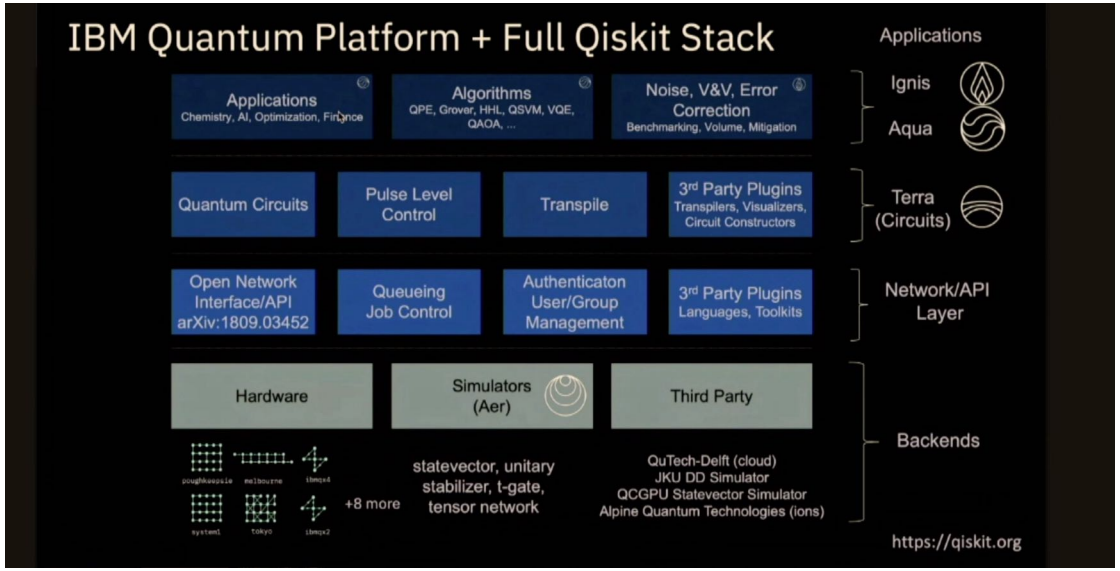
- **The quantum state of superposition is very fragile -** any interaction (heat, radio waves, etc) will collapse the entangled state into a specific value and the **"quantum magic" is gone.**
  - Practically, this means running a quantum computer near absolute zero O(milli-Kelvins)
  - Shielding it from EM interference and vibrations.
- **The "No Cloning Theorem":**
  - It is impossible to create an independent and identical copy of an arbitrary unknown quantum state.
  - Copying values is a fundamental concept in classical numerical algorithms.
  - Practical design consequence: each qubit requires a unique, independent set of I/O channels.
- **So lemme get this straight:**
  - Totally isolate the system of N qbits from the outside world.
  - But run O(N) wires in to interact with these N qbits.
  - Solution: stick something that looks like this -> in a cryogenic fluid.
- **Long term, we need to balance the confounding goals of adding more quantum parallelism while maintaining acceptably low error rates.**
  - We need some sort of **quantum error correction** to pull this off.

IBM's Quantum System 1

Programming a Quantum Computer

- **Quantum supremacy**
  - The point at which quantum computers can perform calculations that are not practical on a classical computer.
- **An Example "real" problem:**
  - Finding the minimum of a function.
  - Has applications for example, to training neural networks in machine learning.
- **Algorithms:**
  - Classical Method of Gradient Descent: notoriously finicky - gets stuck at local minima.
  - Quantum Annealing: probability "pools" at the global minimum.
- **Software:**
  - e.g. Qiskit Software Stack (IBM)
  - Tools for building, controlling and simulating quantum circuits

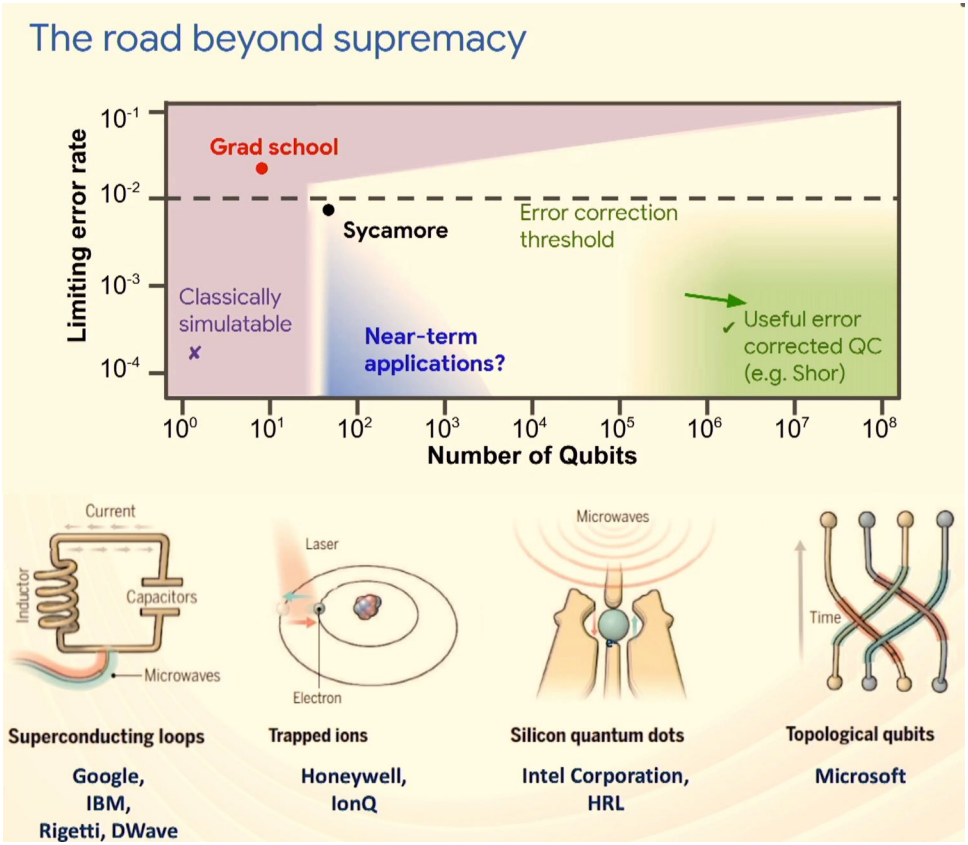




Credit: IBM presentation @ Hot Chips 32 (2020)

Quantum Computing: Path Forward

- **Quantum computing will change HPC & the world**
  - But it will require both scaling to millions of qbits and maintaining sufficiently low error rates.
  - Not clear which qbit technology will win out.
- **We're right back to Charles and Ada's challenge with the Analytical Engine:**
  - Can we build it? (Hardware)
  - What problems can it solve if we do? (Algorithms)
  - How do we make it programmable to solve those problems? (Software)



Quantum Computing Presentation Hot Chips 32 (2020)