Future of Computing: Moore's Law & Its Implications

Introduction to Computer Systems
27th Lecture, Dec. 23, 2024

Instructors:

Class 1: Chen Xiangqun, Liu Xianhua

Class 2: Guan Xuetao

Class 3: Lu Junlin

Moore's Law Origins

April 19, 1965

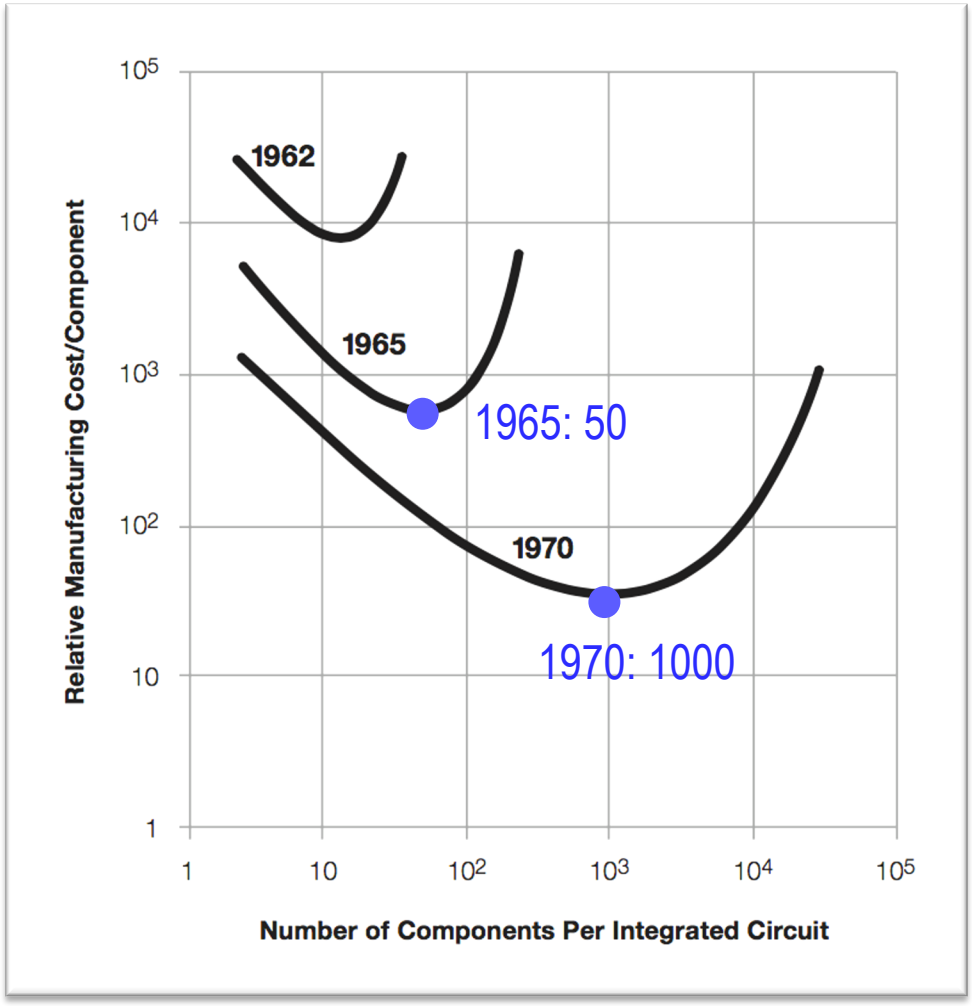
Cramming more components onto integrated circuits

With unit cost falling as the number of components per circuit rises, by 1975 economics may dictate squeezing as many as 65,000 components on a single silicon chip

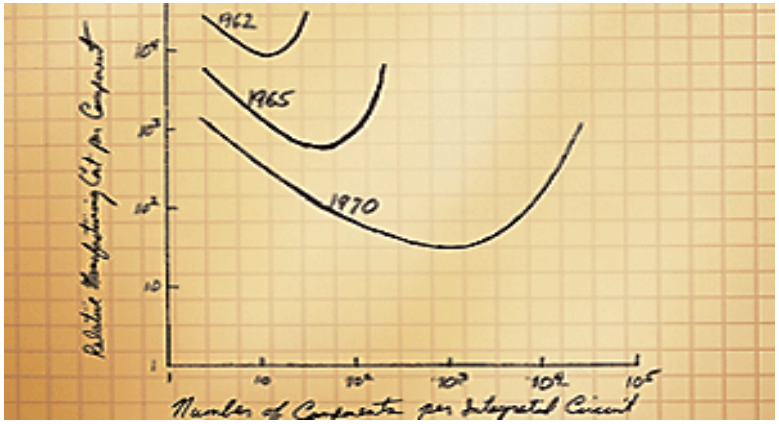
By Gordon E. Moore

Director, Research and Development Laboratories, Fairchild Semiconductor division of Fairchild Camera and Instrument Corp.

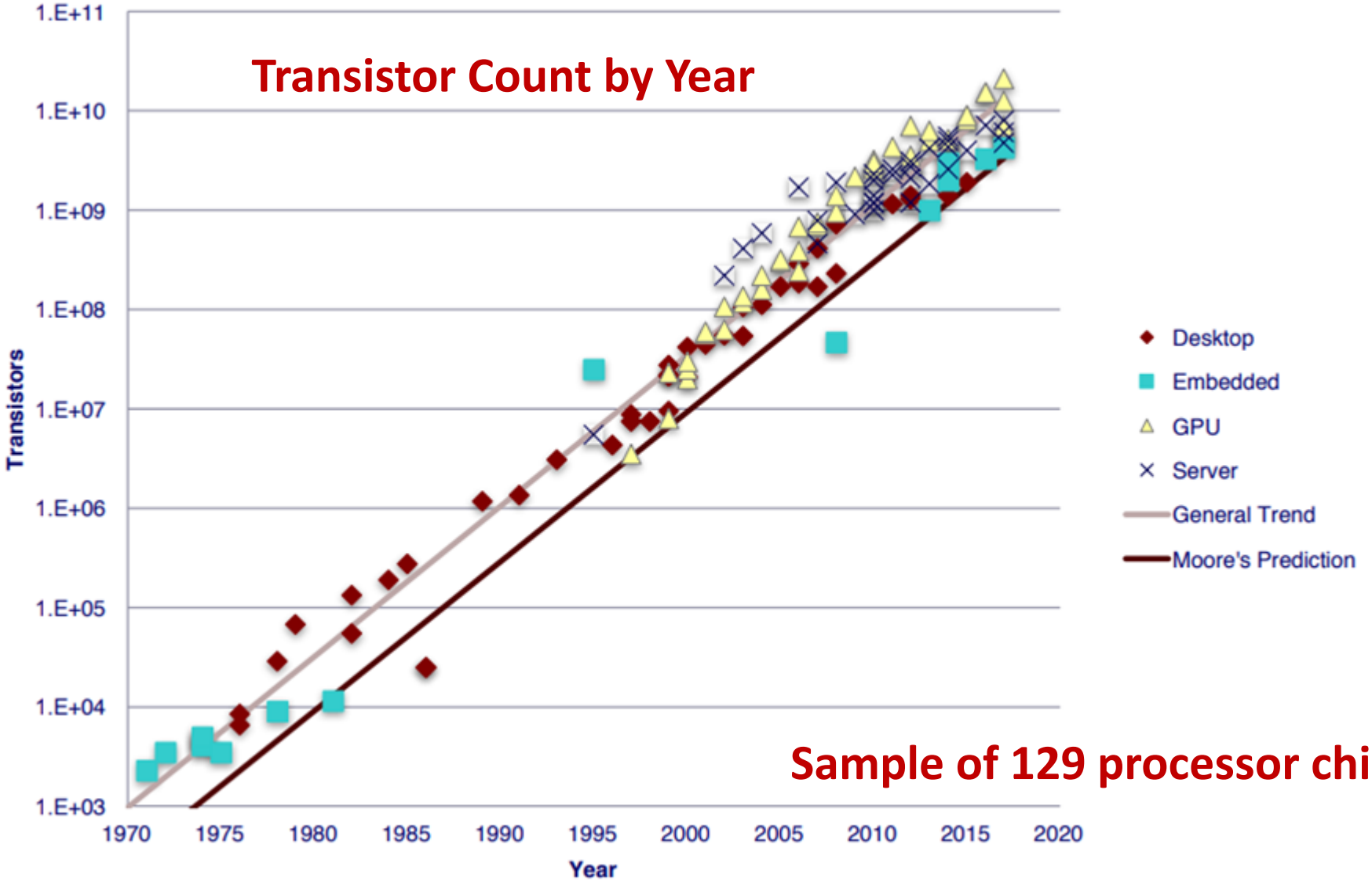
Moore's Law Origins



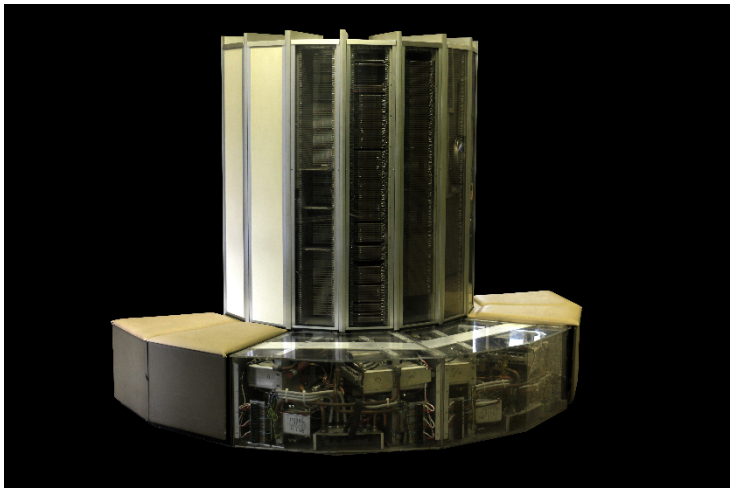
- **Moore's Thesis**
  - Minimize price per device
  - Optimum number of devices / chip increasing 2x / year
- **Later**
  - 2x / 2 years
  - "Moore's Prediction"



Moore's Law: 50 Years



What Moore's Law Has Meant



- **1976 Cray 1**
  - 250 M Ops/second
  - ~170,000 chips
  - 0.5B transistors
  - 5,000 kg, 115 KW
  - $9M
  - 80 manufactured
- **202x Huawei Mate Pro Series**
  - > 10 B Ops/second
  - 8 CPU + 20 GPU + 2 NPU
  - > 15B transistors
  - ~ 200g, < 5 W
  - ~ $800
  - > 14 million sold

What Moore's Law Has Meant

- 1965 Consumer Product

transistor radio

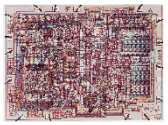
- 2020 Consumer Product

HiSilicon Kirin 9000 with 15.3B transistors

Visualizing Moore's Law to Date

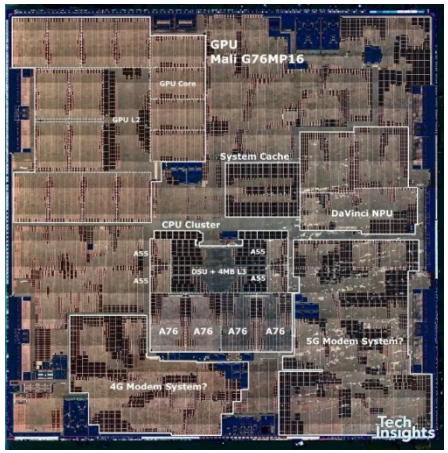
If transistors were the size of a grain of sand

Intel 4004
1970
2,300 transistors



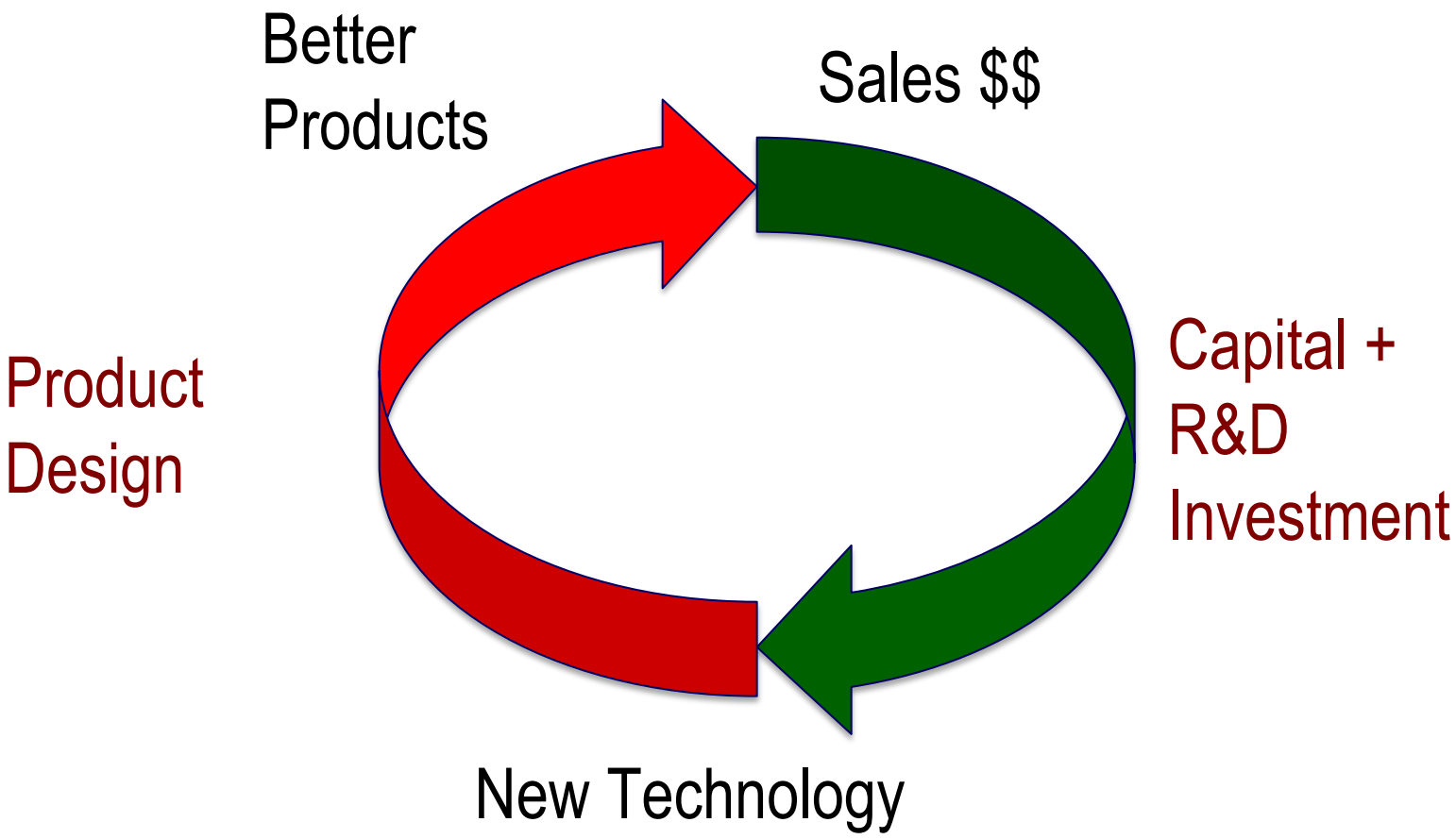
0.1 g

HiSilicon S9000
2020
15.3B transistors



~360 kg

Moore's Law Economics



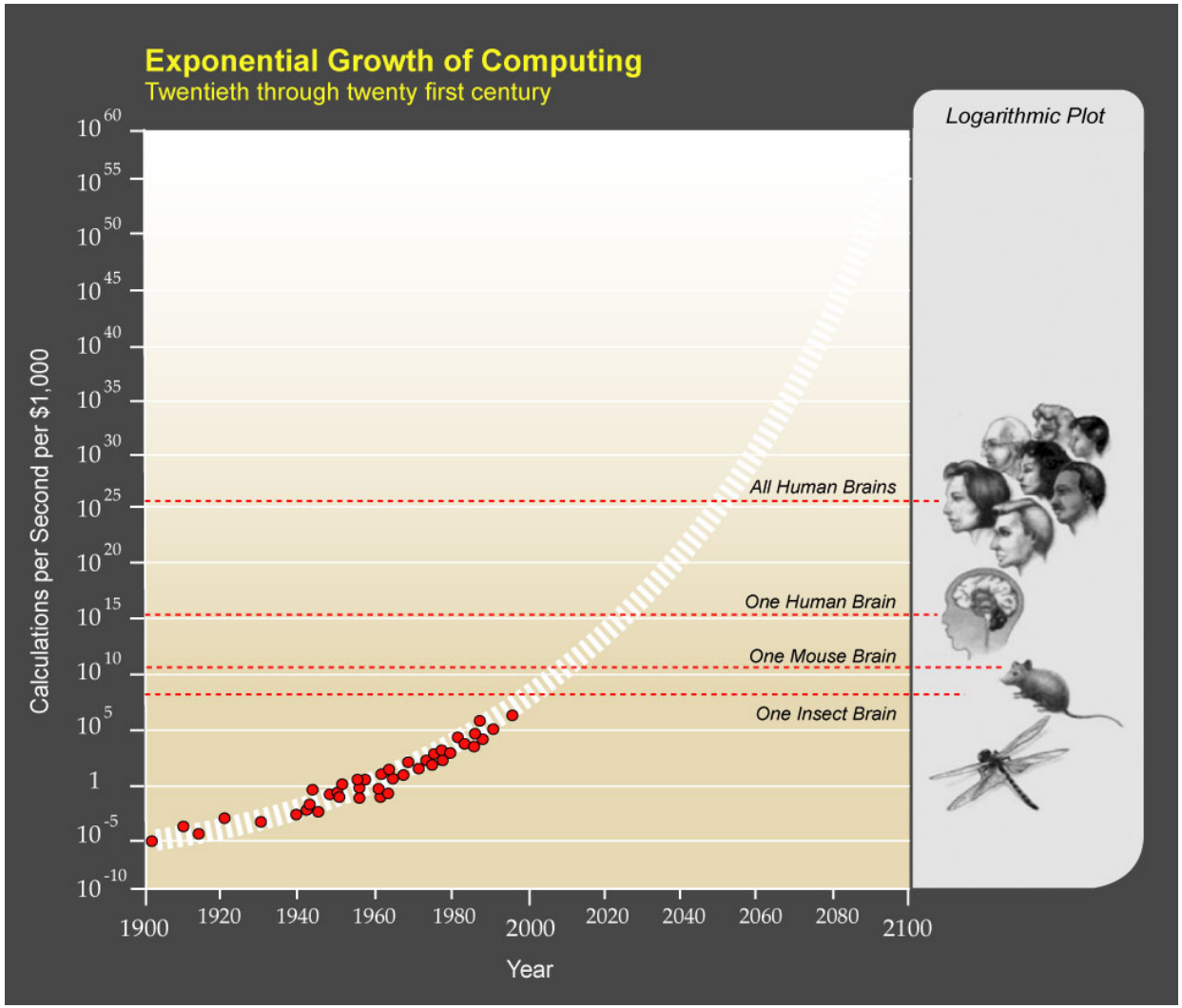
Consumer products sustain the $300B semiconductor industry

What Moore's Law Has Meant

22 generations of iPhone since 2007

- iPhone: June 29, 2007
- iPhone 3G: July 11, 2008
- iPhone 3GS: June 19, 2009
- iPhone 4: June 24, 2010
- iPhone 4S: October 14, 2011
- iPhone 5: September 21, 2012
- iPhone 5S & 5C: September 20, 2013
- iPhone 6 & 6 Plus: September 19, 2014
- iPhone 6S & 6S Plus: September 19, 2015
- iPhone 8 & 8 Plus: September 22, 2017
- iPhone X: November 3, 2017
- iPhone XS, XS Max: September 21, 2018
- iPhone XR: October 26, 2018
- iPhone 11, Pro, Pro Max: September 20, 2019
- iPhone 12, Mini, Pro, Pro Max: November 13, 2020
- iPhone 13, Mini, Pro, Pro Max: September 24, 2021
- iPhone SE 3 (2022): March 18, 2022

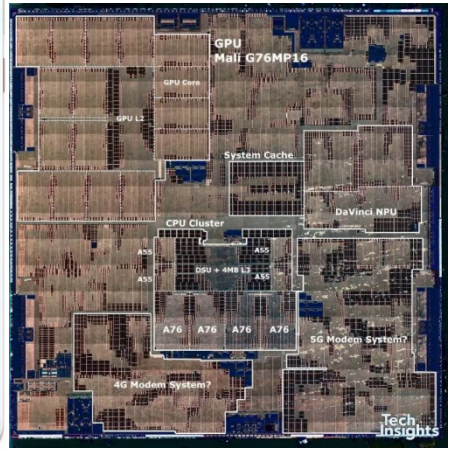
What Moore's Law Could Mean



Kurzweil, *The Singularity is Near*, 2005

What Moore's Law Could Mean

- **202x Consumer Product**



HiSilicon Kirin 9000 with 15.3B transistors

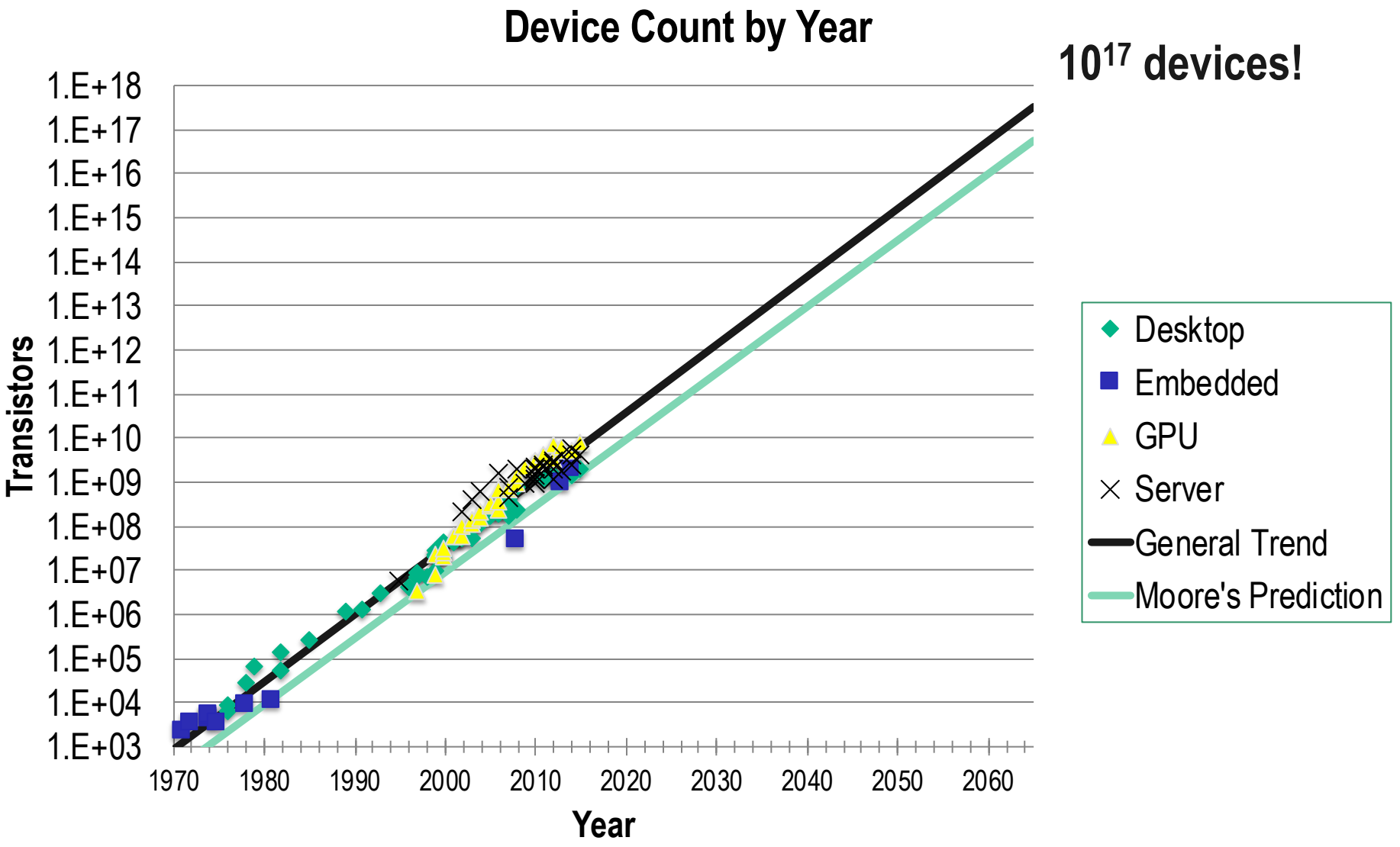
- **2065 Consumer Product**

- Portable
- Low power
- Will drive markets & innovation

Requirements for Future Technology

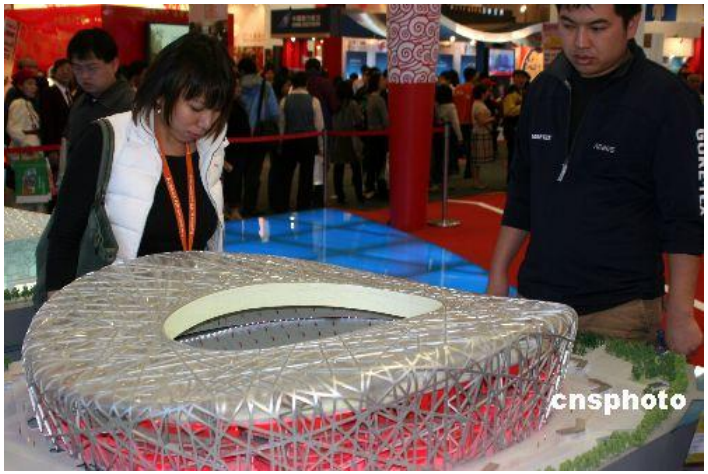
- **Must be suitable for portable, low-power operation**
  - Consumer products
  - Internet of Things components
  - Not cryogenic, not quantum
- **Must be inexpensive to manufacture**
  - Comparable to current semiconductor technology
    - O(1) cost to make chip with O(*N*) devices
- **Need not be based on transistors**
  - Memristors, carbon nanotubes, DNA transcription, ...
  - Possibly new models of computation
  - But, still want lots of devices in an integrated system

Moore's Law: 100 Years



Visualizing 10^{17} Devices

If devices were the size of a grain of sand



$0.1\ m^3$
3.5×10^9 grains

+

$1\ million\ m^3$
0.35×10^{17} grains

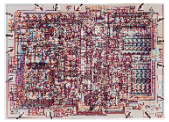
Increasing Transistor Counts

1. **Chips have gotten bigger**
   - 1 area doubling / 10 years
2. **Transistors have gotten smaller**
   - 4 density doublings / 10 years

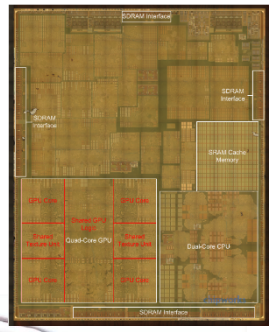
Will these trends continue?

Chips Have Gotten Bigger

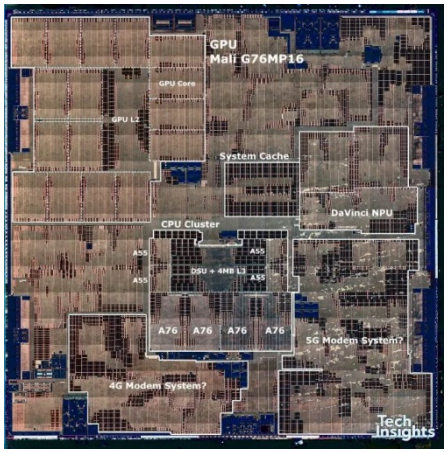
Intel 4004
1970
2,300 transistors
12 mm²



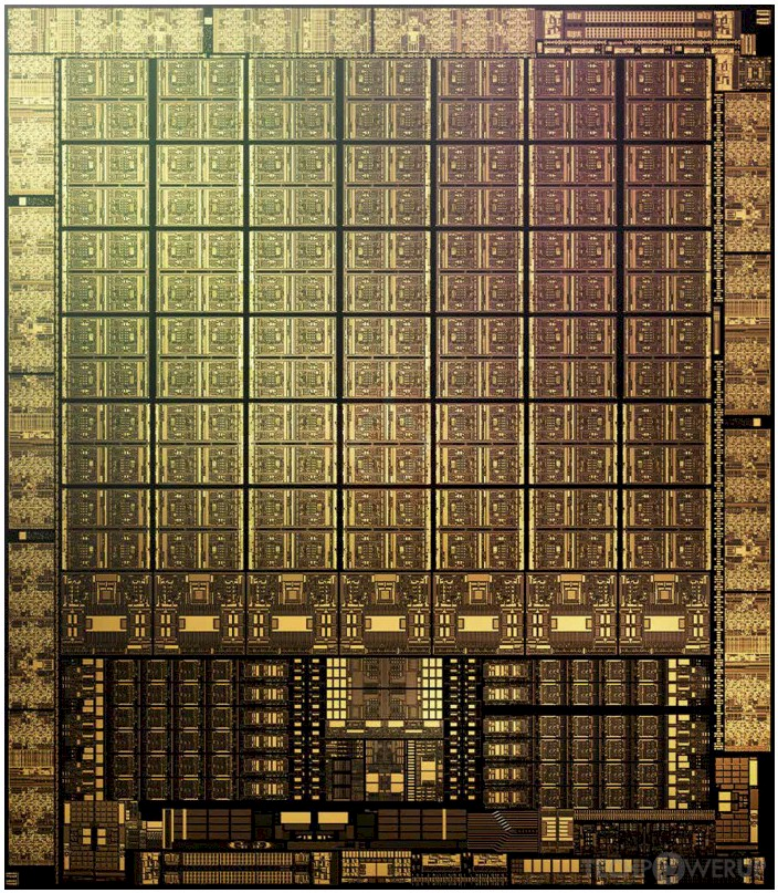
Apple A8
2014
2 B transistors
89 mm²



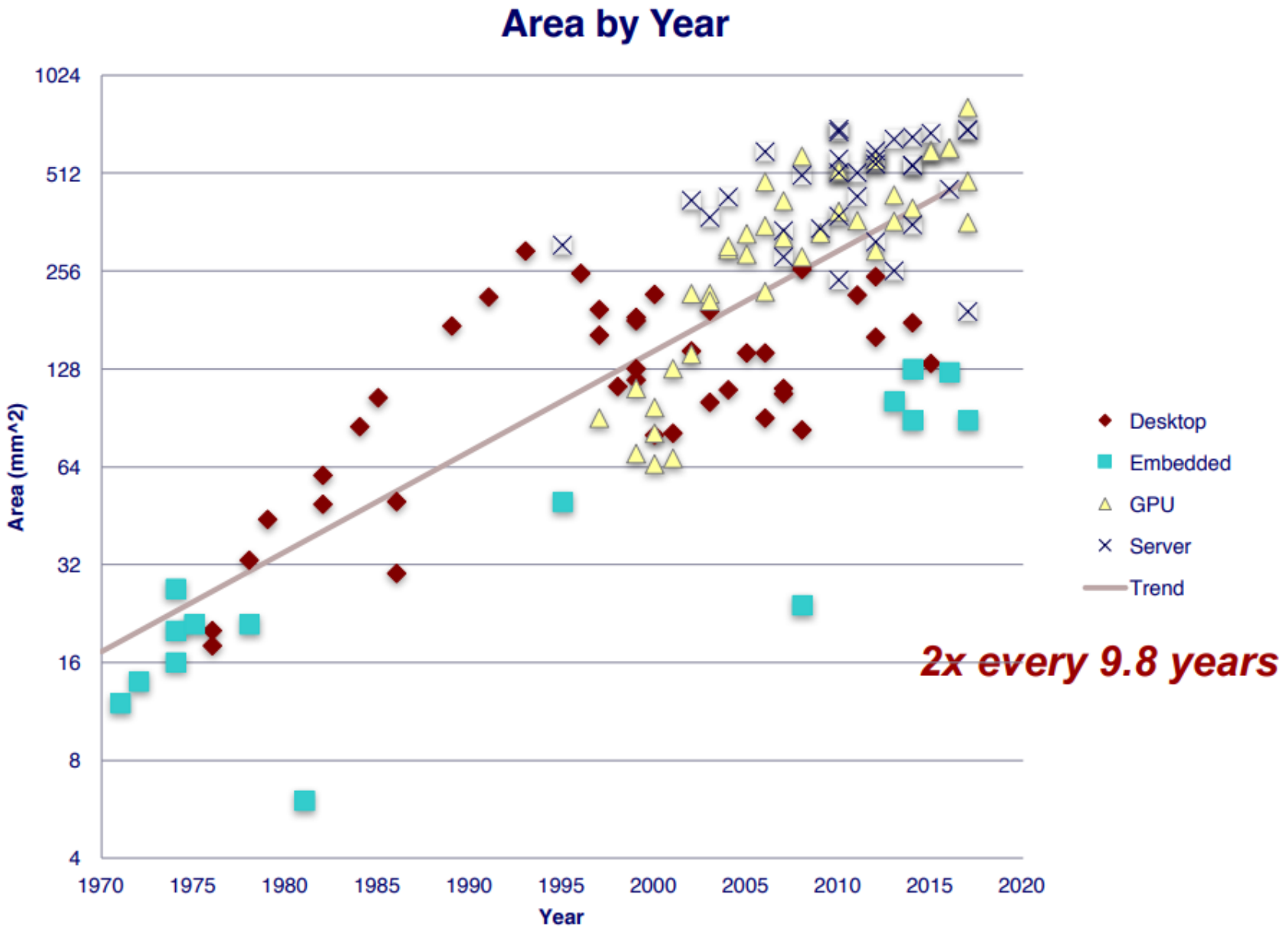
HiSilicon S9000
2020
15 B transistors, 5nm
110 mm²



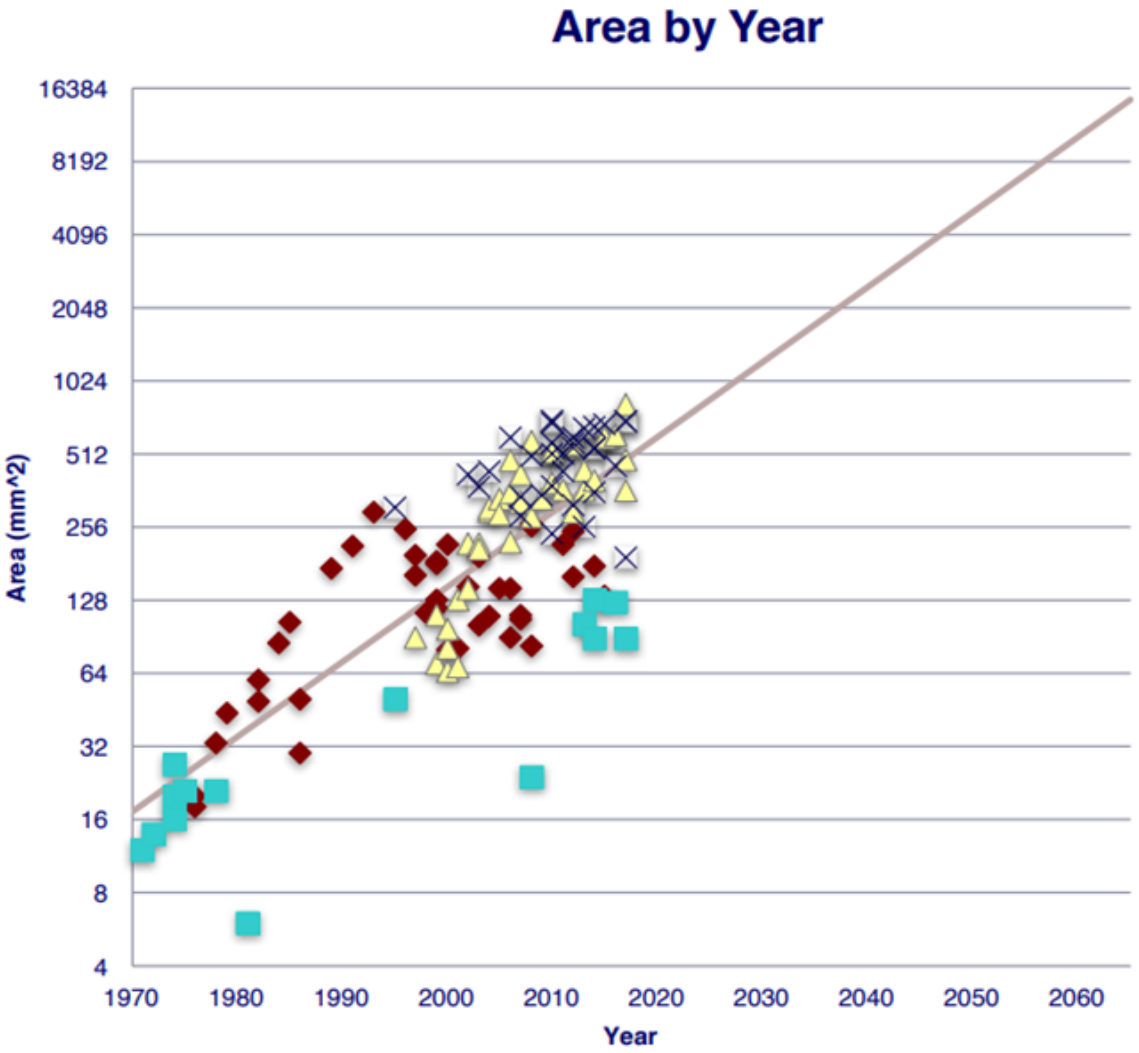
NVIDIA A100
2020
54 B transistors, 7nm
826 mm²



Chip Size Trend



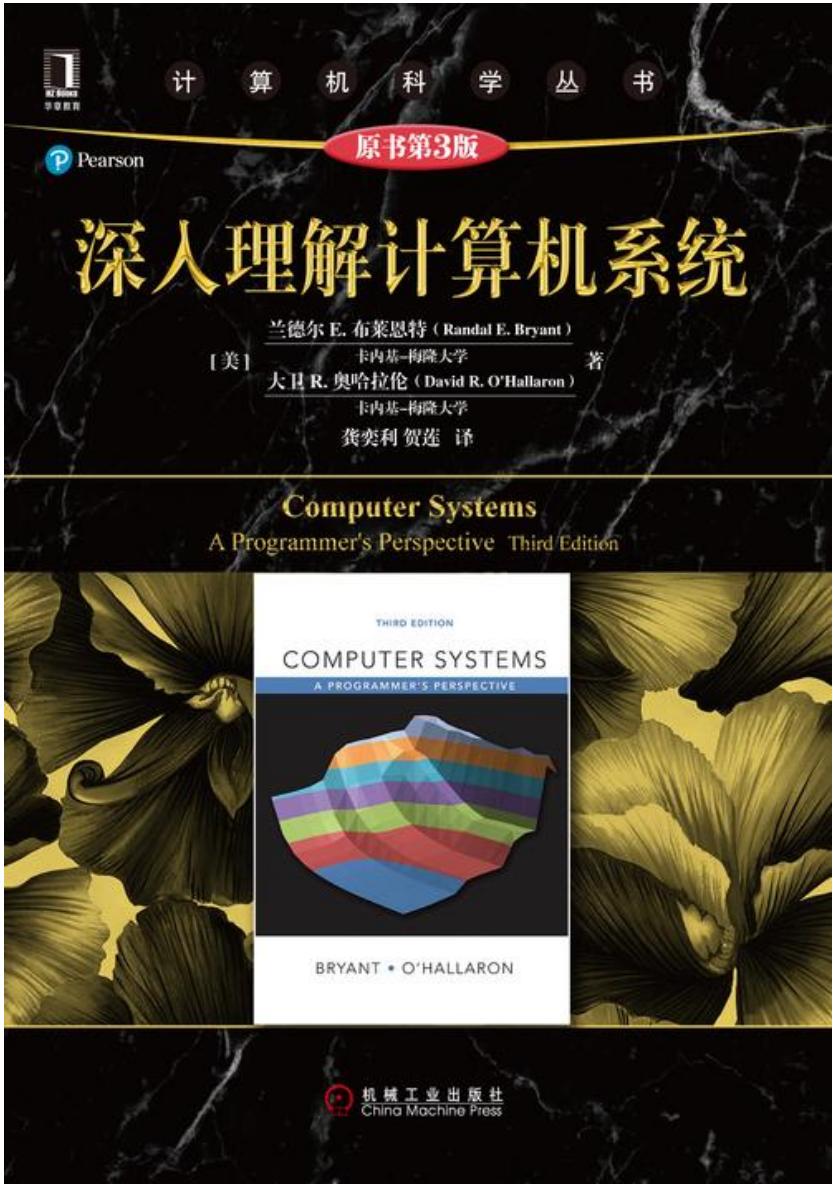
Chip Size Extrapolation



187 cm^2

Extrapolation: The iPhone 101s

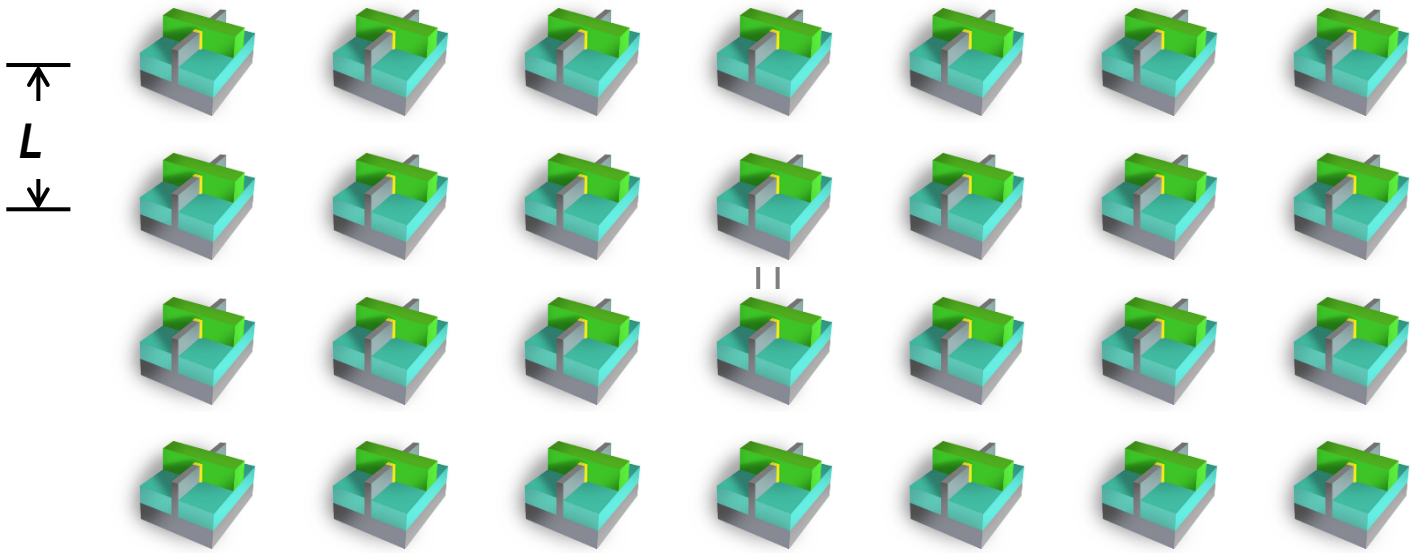
Apple Axxx
2065
10^{17} transistors
187 cm^2



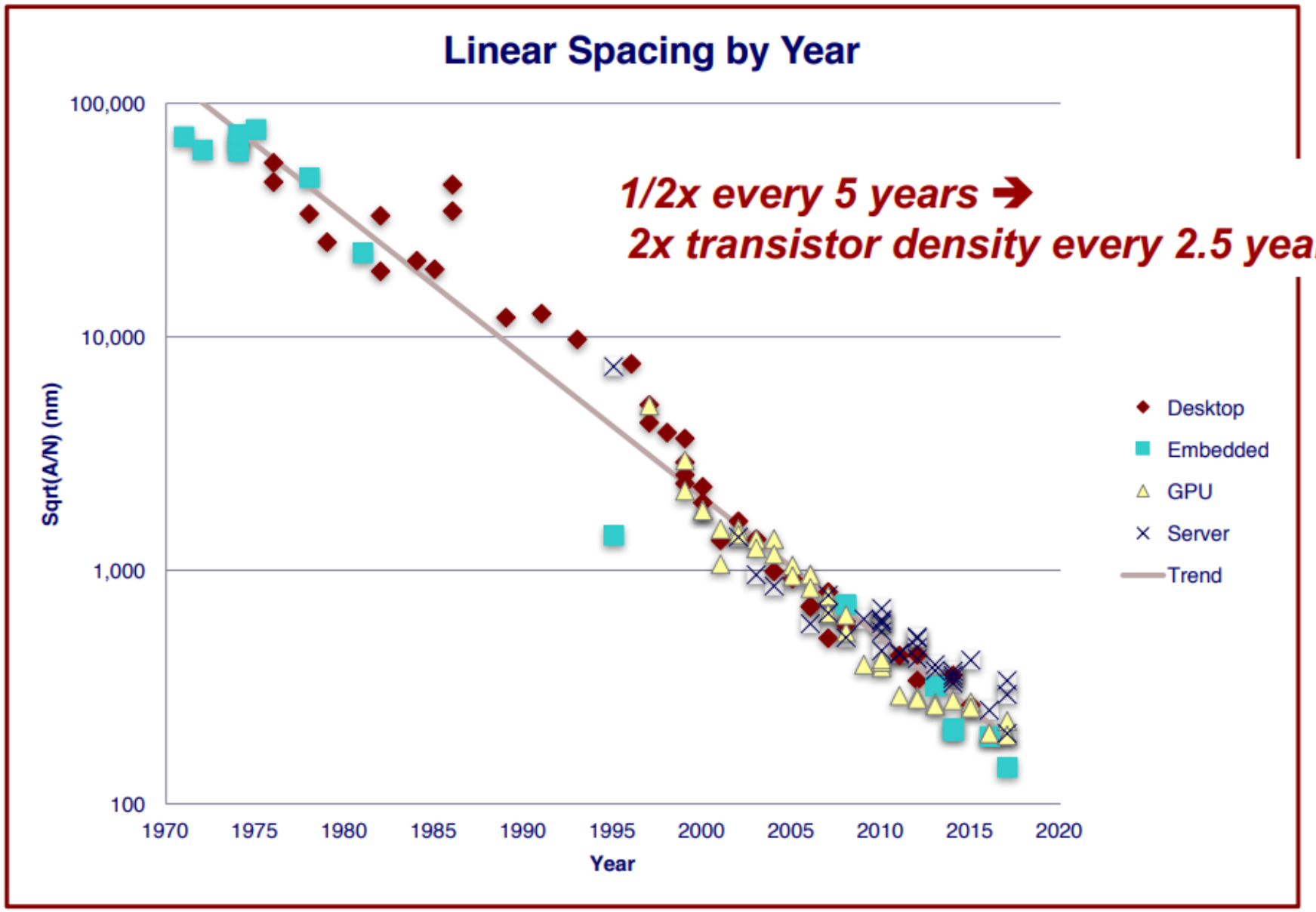
Transistors Have Gotten Smaller

- Area *A*
- *N* devices
- Linear Scale *L*

$$L = \sqrt{A / N}$$

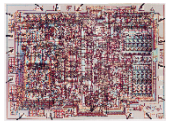


Linear Scaling Trend

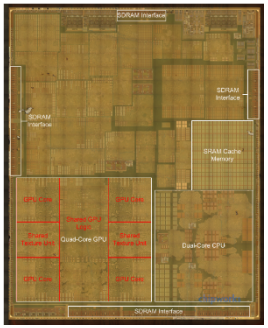


Decreasing Feature Sizes

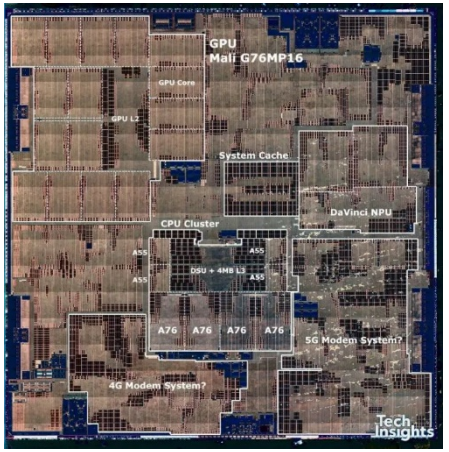
Intel 4004
1970
2,300 transistors
L = 72,000 nm



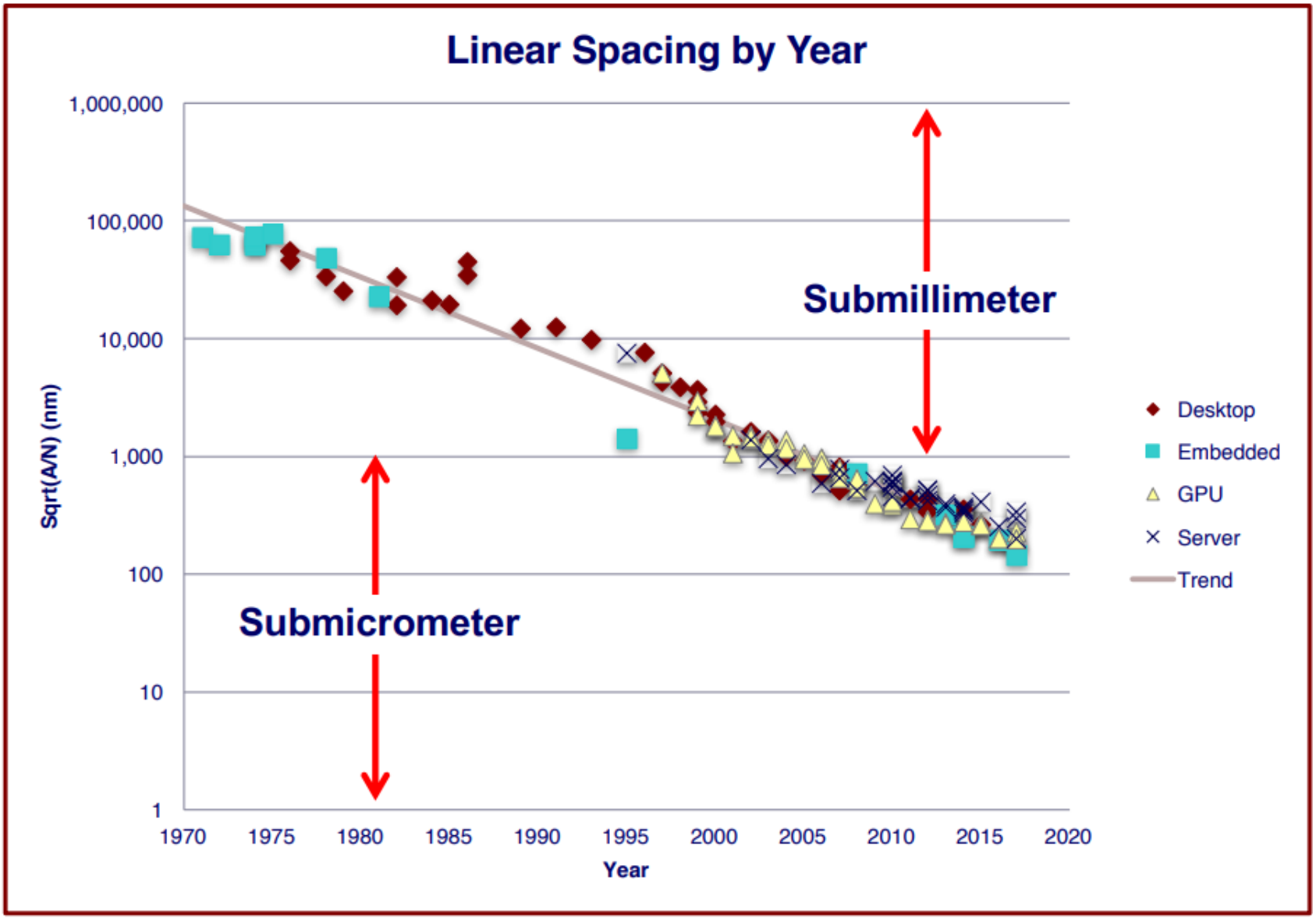
Apple A8
2014
2 B transistors
L = 211 nm



HiSilicon S9000
2020
15.3B transistors
L= 85nm



Linear Scaling Trend

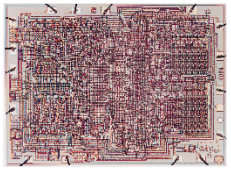


Submillimeter Dimensions

10^{-3} 1 millimeter (mm)

500μm: Length of amoeba

10^{-4}



100μm: A4 paper thickness

72μm: Intel 4004 linear scale/ The diameter of human hair

50μm: Average size of cell in human body

10^{-5}

10μm: Thickness of sheet of plastic food wrap

5μm: Spider silk thickness

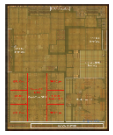
2μm: Thickness of red blood cells

10^{-6} 1 micrometer (μm)

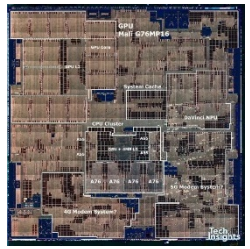
Submicrometer Dimensions

10^{-6} 1 micrometer

400-700nm: Visible light wavelengths



211nm: Apple A8 linear scale



10^{-7}

85nm: HiSilicon Kirin S9000 linear scale

30nm: Minimum cooking oil smoke particle diameter

10^{-8}

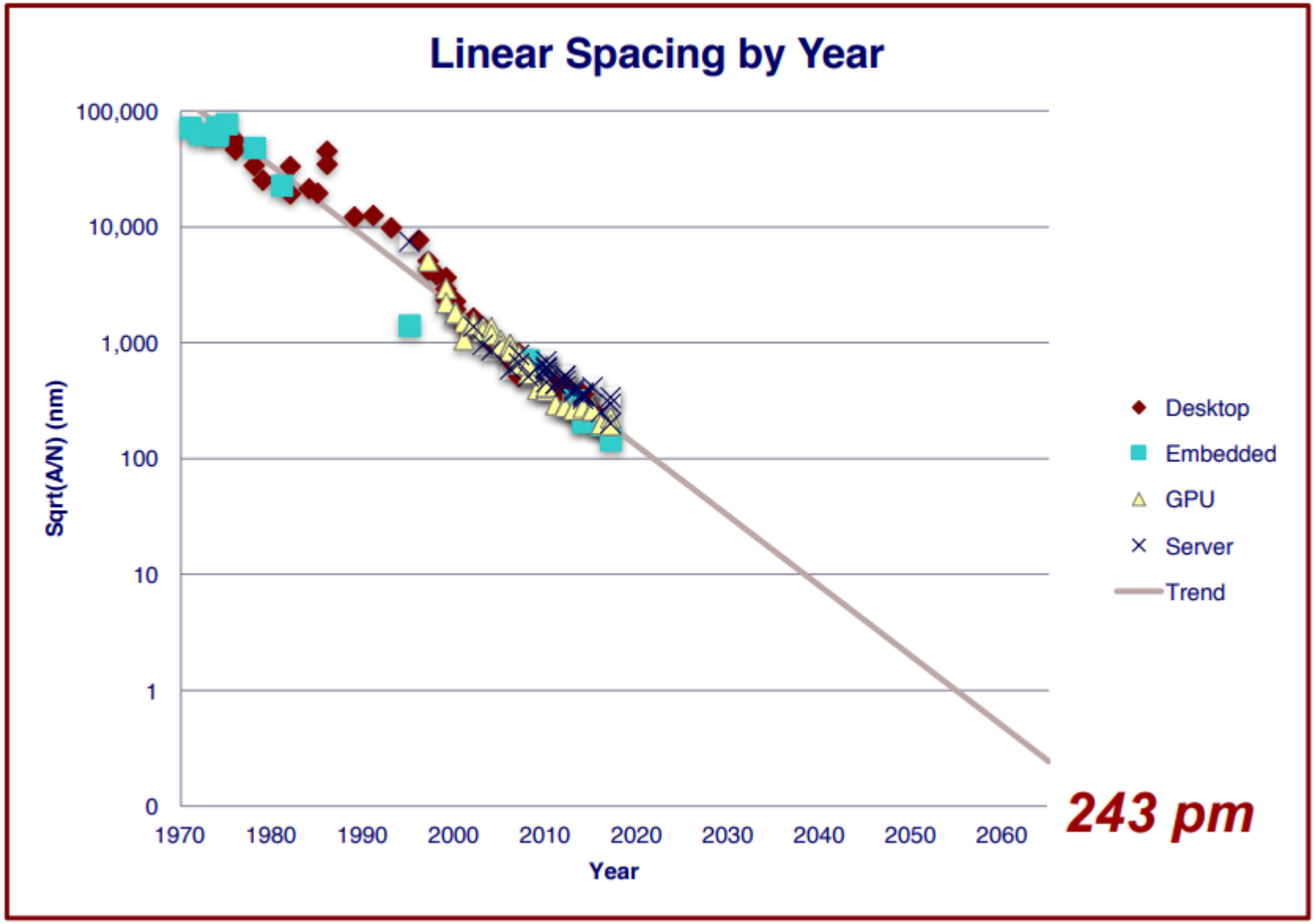
9nm: Cell membrane thickness

2nm: DNA helix diameter

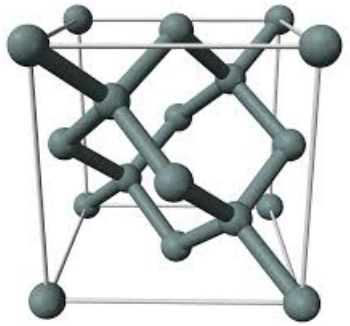
10^{-9} 1 nanometer (nm)

1nm: Carbon nanotube diameter

Linear Scaling Extrapolation



Subnanometer Dimensions



10^{-9} 1 nanometer (nm)

1nm: Carbon nanotube diameter

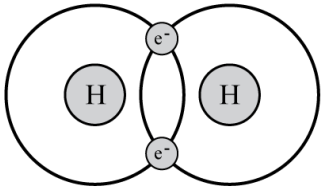
543pm: Silicon crystal lattice spacing

243pm: 2065 linear scale projection

10^{-10}

74pm: Spacing between atoms in hydrogen molecule

53pm: Electron-proton spacing in hydrogen (Bohr radius)



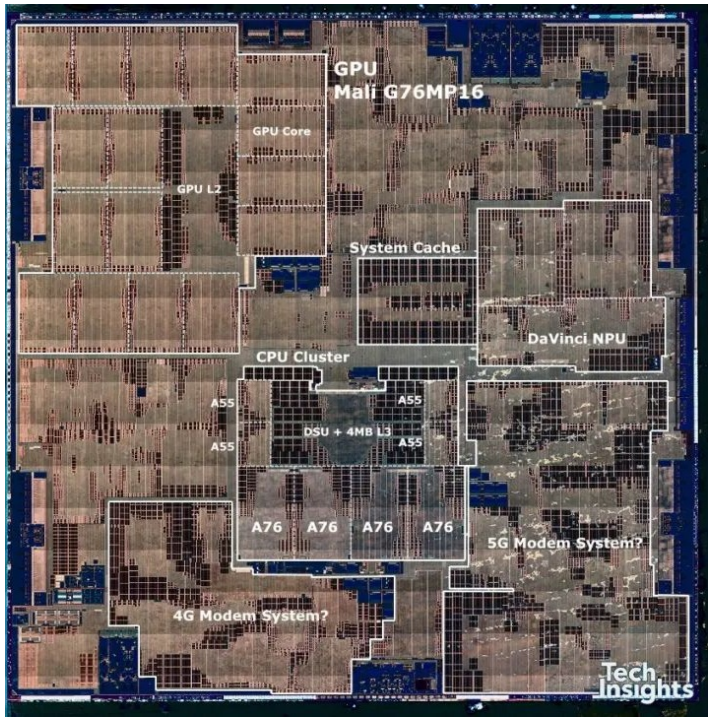
10^{-11}

2.4pm: Electron wavelength (Compton wavelength)

10^{-12} 1 picometer (pm)

Reaching 2065 Goal

- **Target**
  - 10^{17} devices
  - 400 mm^2
  - L = 63 pm

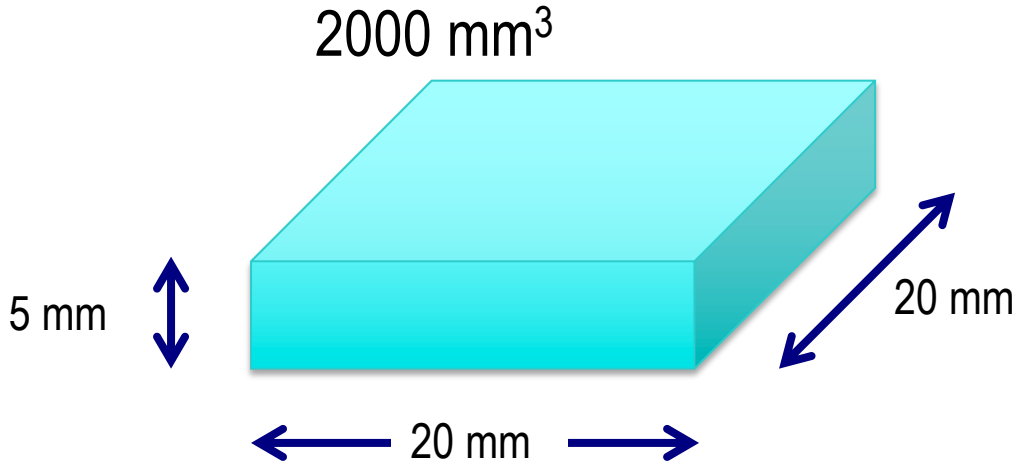


- **Is this possible?**

NO!

Not with 2-D fabrication

Fabricating in 3 Dimensions



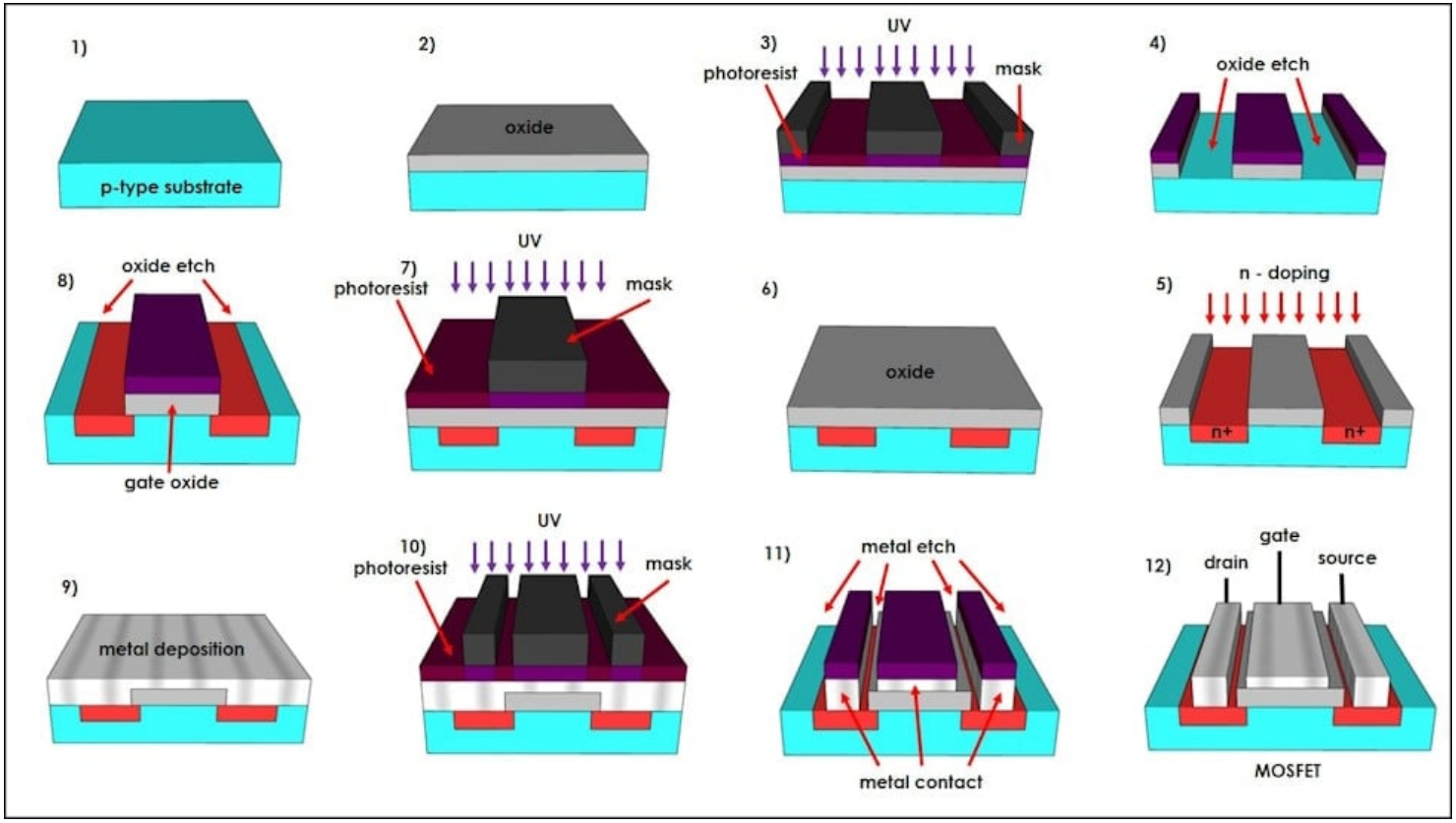
- **Parameters**
  - 10^{17} devices
  - 100,000 logical layers
    - Each 50 nm thick
    - ~1,000,000 physical layers
      - To provide wiring and isolation
  - *L* = 20 nm
    - 10x smaller than today

≈ 2000 mm3

3D Fabrication Challenges

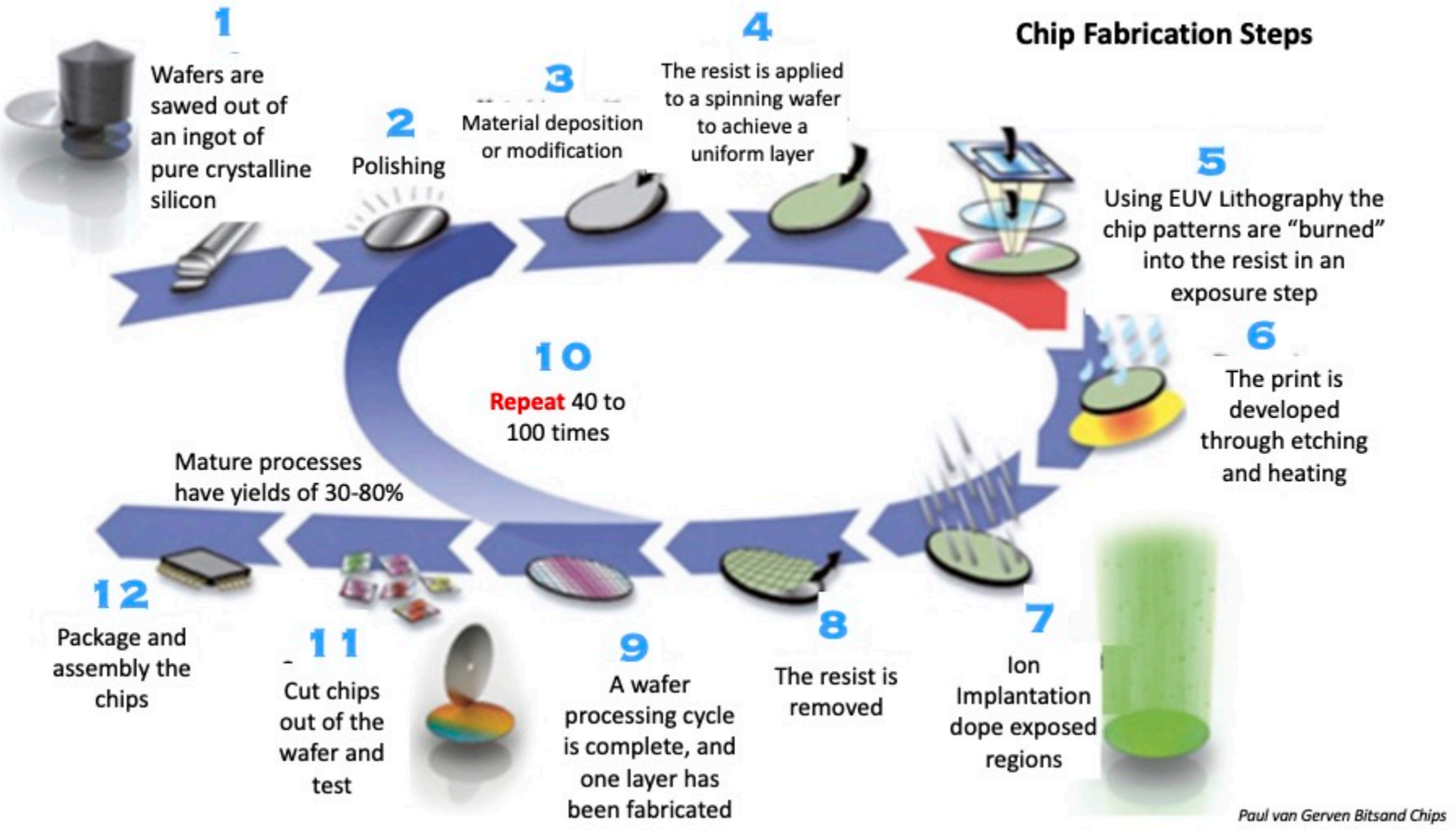
- **Yield**
  - How to avoid or tolerate flaws
- **Cost**
  - High cost of lithography
- **Power**
  - Keep power consumption within acceptable limits
  - Limited energy available
  - Limited ability to dissipate heat

Photolithography

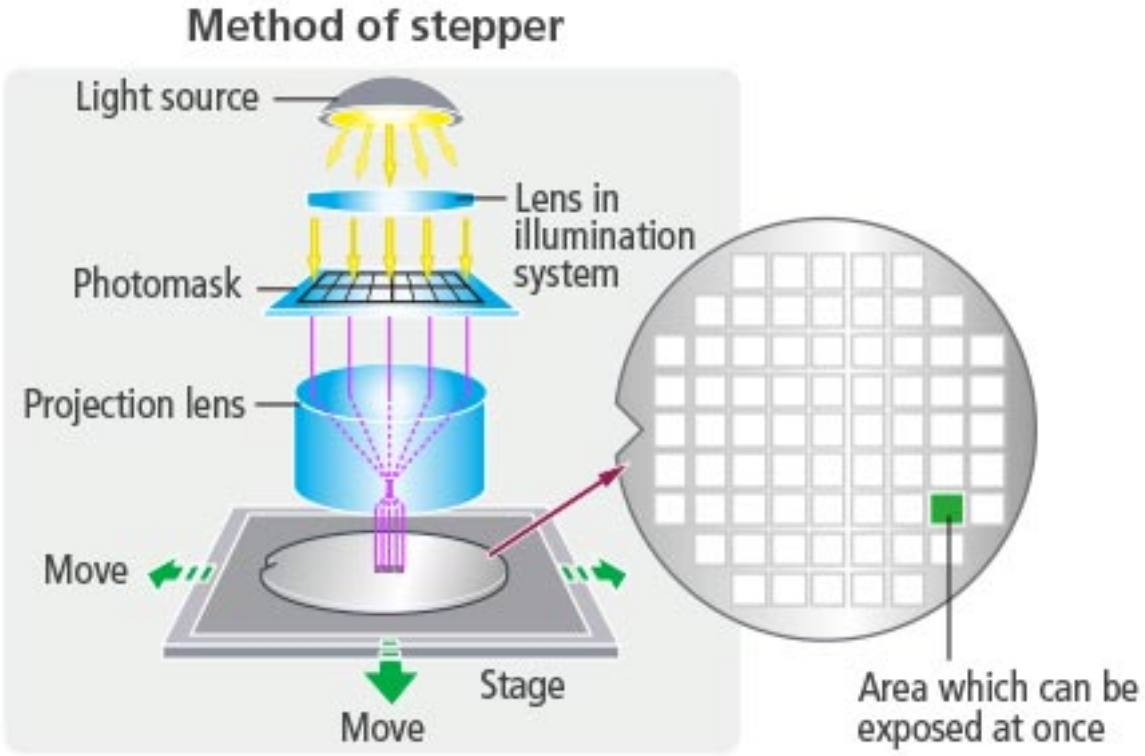


- Pattern entire chip in one step
- Modern chips require ~60 lithography steps
- Fabricate *N* transistor system with O(1) steps

Semiconductor Fabrication Process



Fabrication Costs

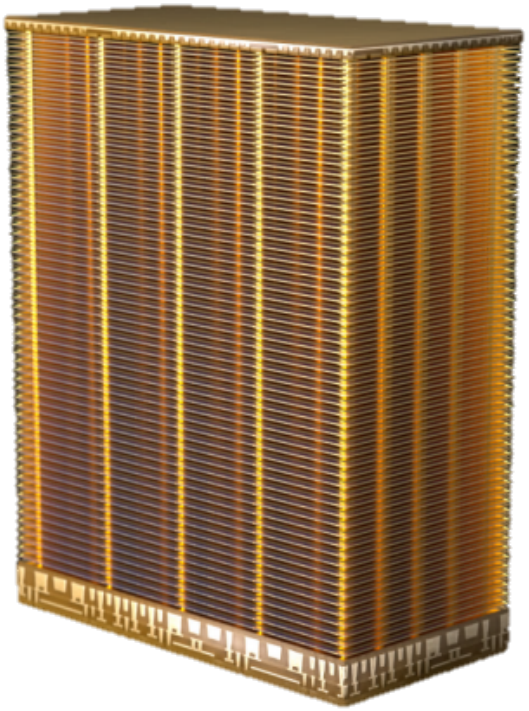


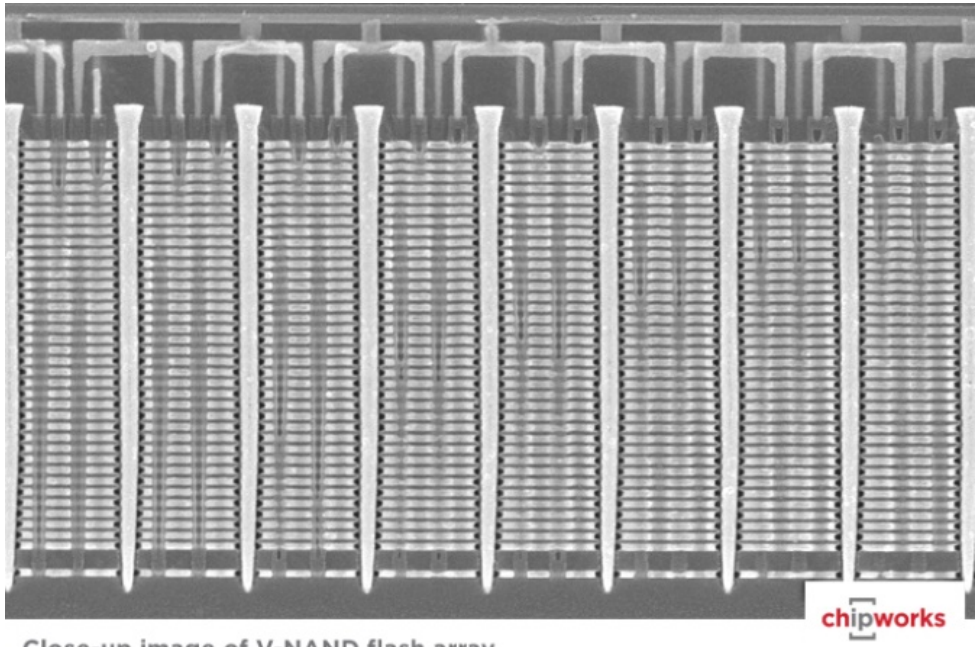
- **Stepper**
  - Most expensive equipment in fabrication facility
  - Rate limiting process step
    - 18s / wafer
  - Expose 858 mm^2 per step
    - 1.2% of chip area

Fabrication Economics

- **Currently**
  - Fixed number of lithography steps
  - Manufacturing cost $10–$20 / chip
    - Including amortization of facility
- **Fabricating 1,000,000 physical layers**
  - Cannot do lithography on every step
- **Options**
  - Chemical self assembly
    - Devices generate themselves via chemical processes
  - Pattern multiple layers at once

Samsung V-Nand Flash Example

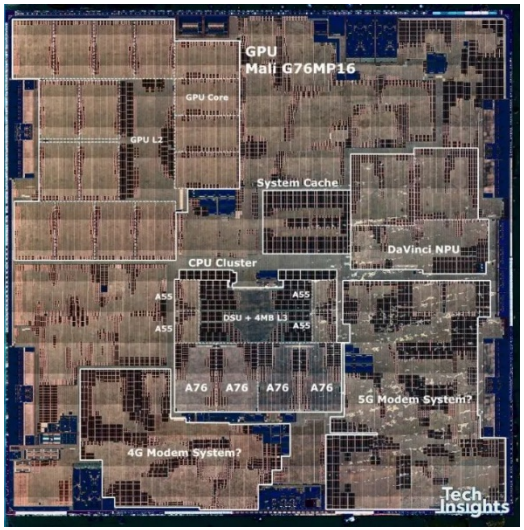




Close-up image of V-NAND flash array

- Build up layers of unpatterned material
- Then use lithography to slice, drill, etch, and deposit material across all layers
- Over 30 total masking steps
- 300+ layers of memory cells (SK Hynix，2023.08)
- Exploits particular structure of flash memory circuits

Meeting Power Constraints



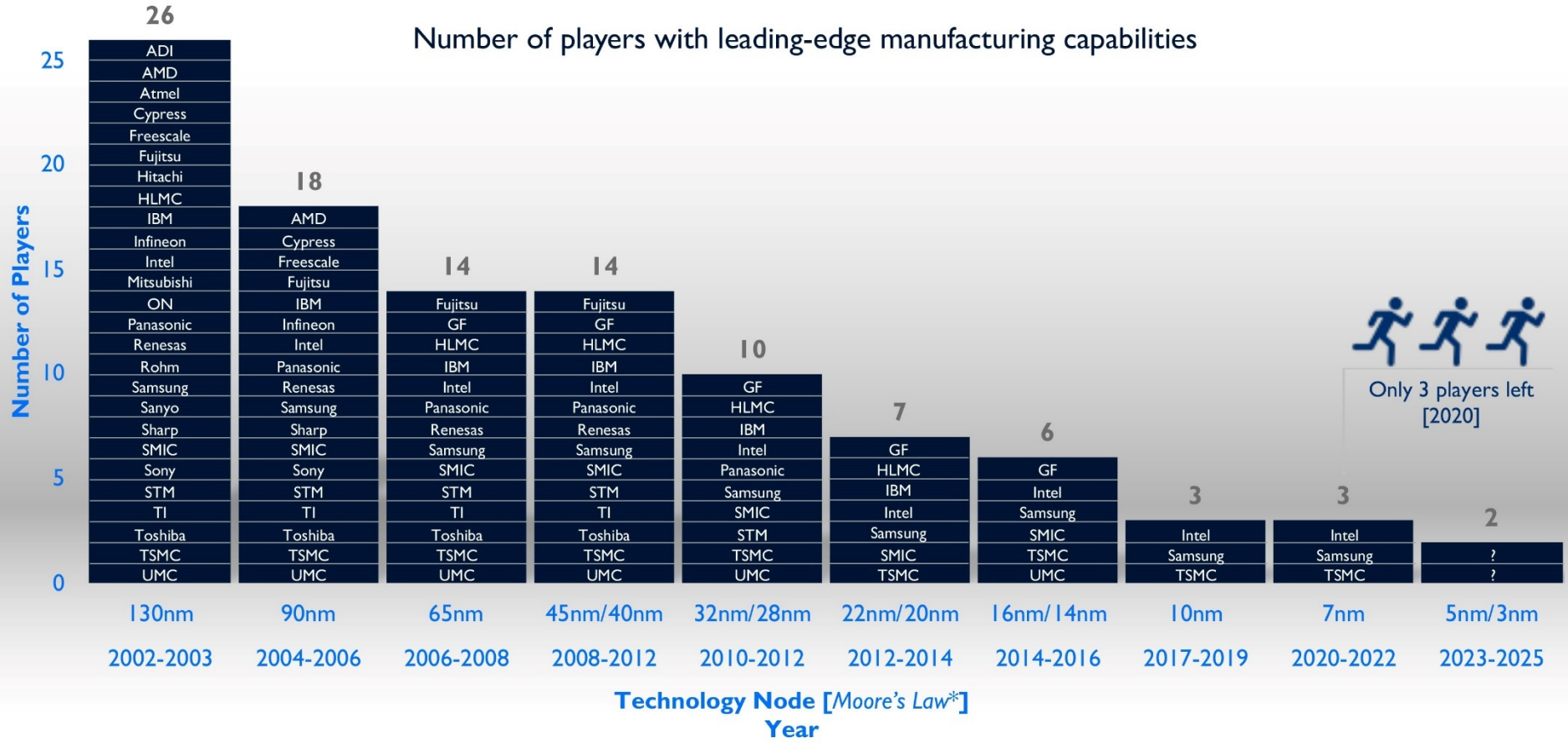
- 15.3 B transistors
- 2.62 GHz operation
- 1—5 W

- 64 B neurons
- 100 Hz operation
- 15—25 W
  - Liquid cooling
  - Up to 25% body's total energy consumption

Can we increase number of devices by 500,000x without increasing power requirement?

Challenges to Moore's Law: Economic

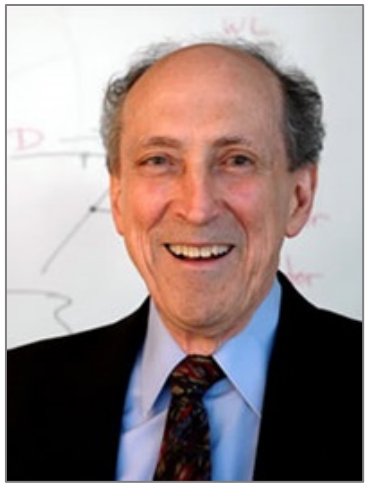
- **Growing Capital Costs**
  - State of art fab line > $20B
  - Must have very high volumes to amortize investment
  - Has led to major consolidations



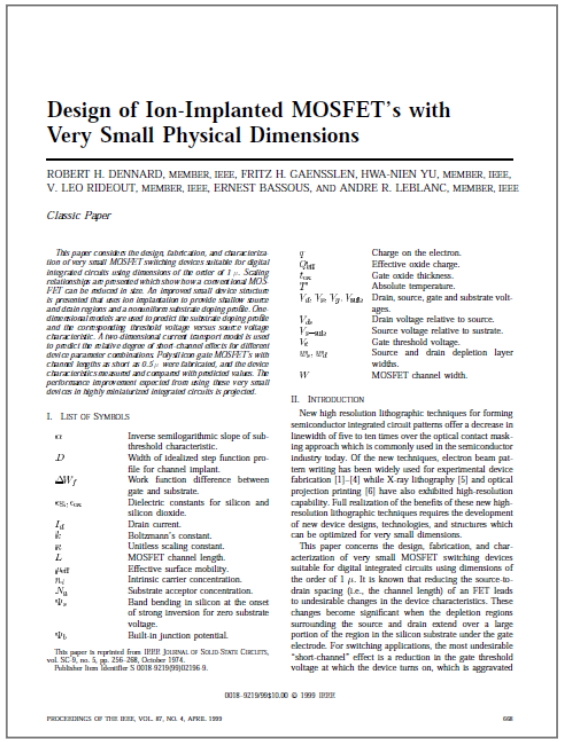
Dennard Scaling

Due to Robert Dennard, IBM, 1974

Quantifies benefits of Moore's Law



- **How to shrink an IC Process**
  - Reduce horizontal and vertical dimensions by *k*
  - Reduce voltage by *k*
- **Outcomes**
  - Devices / chip increase by k^2
  - Clock frequency increases by *k*
  - Power / chip constant
- **Significance**
  - Increased capacity and performance
  - No increase in power



Design of Ion-Implanted MOSFET's with Very Small Physical Dimensions

ROBERT H. DENNARD, MEMBER, IEEE, FRITZ H. GAENSSLEN, HWA-NIEN YU, MEMBER, IEEE, V. LEO RIDEOUT, MEMBER, IEEE, ERNEST BASSOUS, AND ANDRE R. LEBLANC, MEMBER, IEEE

Classic Paper

This paper considers the design, fabrication, and characterization of very small MOSFET switching devices suitable for digital integrated circuits using dimensions of the order of 1 μ. Scaling relationships are presented which show how a conventional MOSFET can be reduced in size. An improved small device structure is presented that uses ion implantation to provide shallow source and drain regions and a nonuniform substrate doping profile. One-dimensional models are used to predict the substrate doping profile and the corresponding threshold voltage versus source voltage characteristic. A two-dimensional current transport model is used to predict the relative degree of short-channel effects for different device parameter combinations. Polysilicon gate MOSFET's with channel lengths as short as 0.5 μ were fabricated, and the device characteristics measured and compared with predicted values. The performance improvement expected from using these very small devices in highly miniaturized integrated circuits is projected.

I. List of Symbols

$\kappa\varepsilon$	Inverse semilogarithmic slope of subthreshold characteristic.
D	Width of idealized step function profile for channel implant.
ΔW_f	Work function difference between gate and substrate.
κ_{Si}, κ_{ox}	Dielectric constants for silicon and silicon dioxide.
I_d	Drain current.
k	Boltzmann's constant.
κ	Unitless scaling constant.
L	MOSFET channel length.
μ_{eff}	Effective surface mobility.
n_i	Intrinsic carrier concentration.
N_a	Substrate acceptor concentration.
Ψ_s	Band bending in silicon at the onset of strong inversion for zero substrate voltage.
Ψ_b	Built-in junction potential.
q	Charge on the electron.
Q_{ss}	Effective oxide charge.
t_{ox}	Gate oxide thickness.
T	Absolute temperature.
V_d, V_s, V_g, V_{sub}	Drain, source, gate and substrate voltages.
V_{ds}	Drain voltage relative to source.
$V_{s\to sub}$	Source voltage relative to sustrate.
V_t	Gate threshold voltage.
w_s, w_d	Source and drain depletion layer widths.
W	MOSFET channel width.

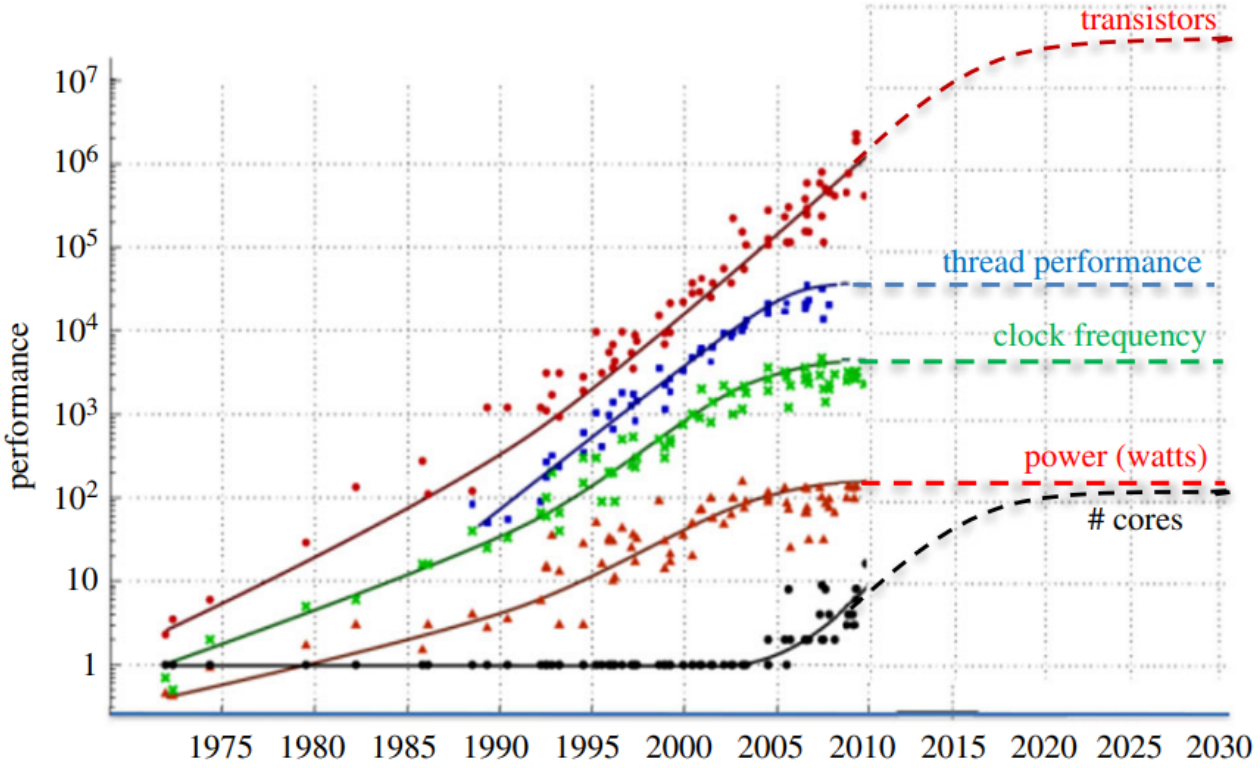
II. Introduction

New high resolution lithographic techniques for forming semiconductor integrated circuit patterns offer a decrease in linewidth of five to ten times over the optical contact masking approach which is commonly used in the semiconductor industry today. Of the new techniques, electron beam pattern writing has been widely used for experimental device fabrication [1]–[4] while X-ray lithography [5] and optical projection printing [6] have also exhibited high-resolution capability. Full realization of the benefits of these new high-resolution lithographic techniques requires the development of new device designs, technologies, and structures which can be optimized for very small dimensions.

This paper concerns the design, fabrication, and characterization of very small MOSFET switching devices suitable for digital integrated circuits using dimensions of the order of 1 μ. It is known that reducing the source-to-drain spacing (i.e., the channel length) of an FET leads to undesirable changes in the device characteristics. These changes become significant when the depletion regions surrounding the source and drain extend over a large portion of the region in the silicon substrate under the gate electrode. For switching applications, the most undesirable "short-channel" effect is a reduction in the gate threshold voltage at which the device turns on, which is aggravated

This paper is reprinted from IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. SC-9, no. 5, pp. 256–268, October 1974.
Publisher Item Identifier S 0018-9219(99)02196-9.

End of Dennard Scaling



- **What Happened?**
  - Can't drop voltage below ~1V
  - Reached limit of power / chip in 2004
  - More logic on chip (Moore's Law), but can't make them run faster
    - Response has been to increase cores / chip

Some Thoughts about Technology

- **Compared to future, past 50 years will seem fairly straightforward**
  - 50 years of using photolithography to pattern transistors on two-dimensional surface
- **Questions about future integrated systems**
  - Can we build them?
  - What will be the technology?
  - Are they commercially viable?
  - Can we keep power consumption low?
  - What will we do with them?
  - How will we program / customize them?

Future of Computing

HIGH-PERFORMANCE COMPUTING

Comparing Two Large-Scale Systems

- **Oakridge Titan**

- Monolithic supercomputer (4th fastest in world @2017, rank 63th now)
- Designed for compute-intensive applications

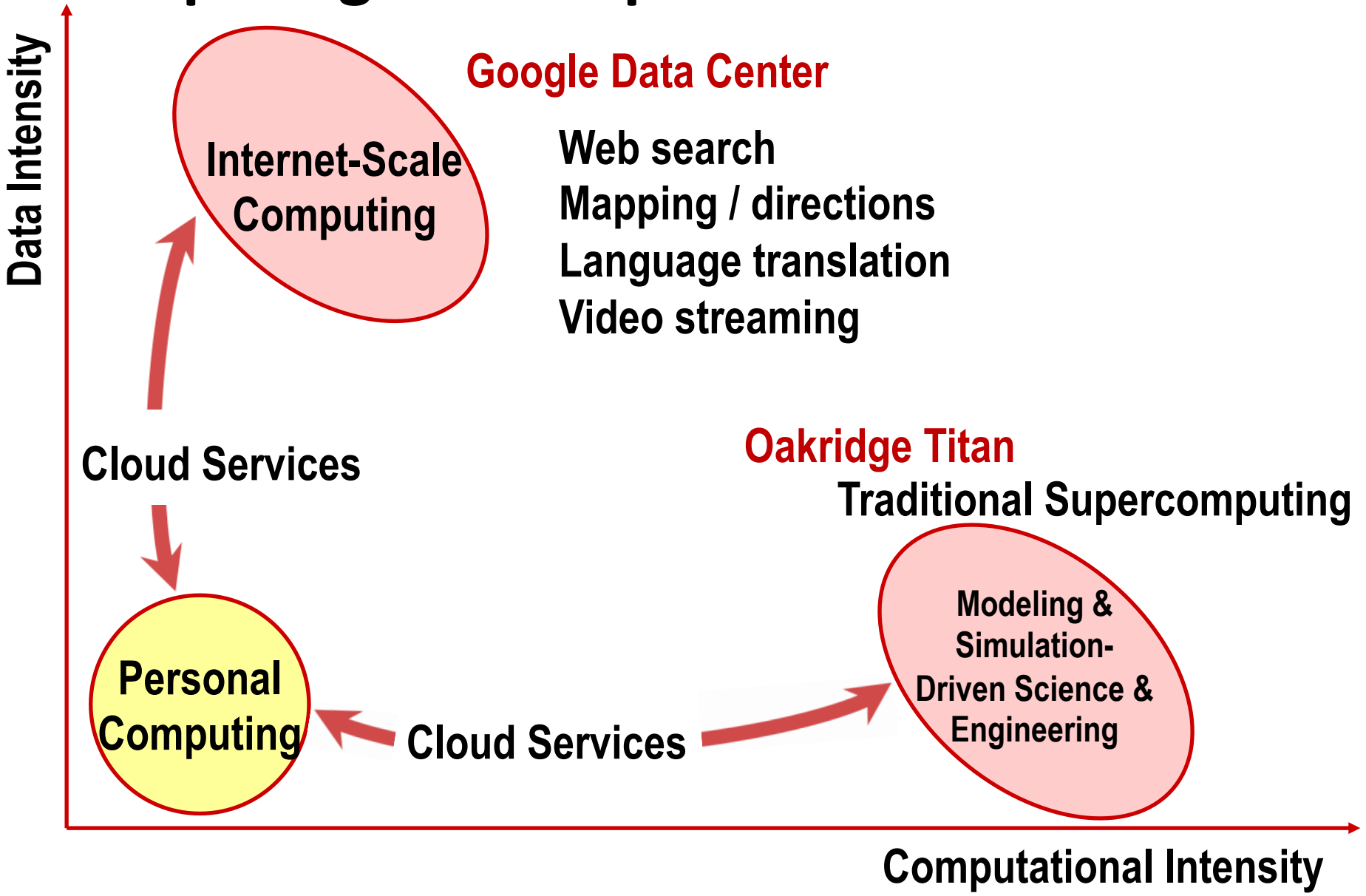
- **Google Data Center**

- Servers to support millions of customers
- Designed for data collection, storage, and analysis

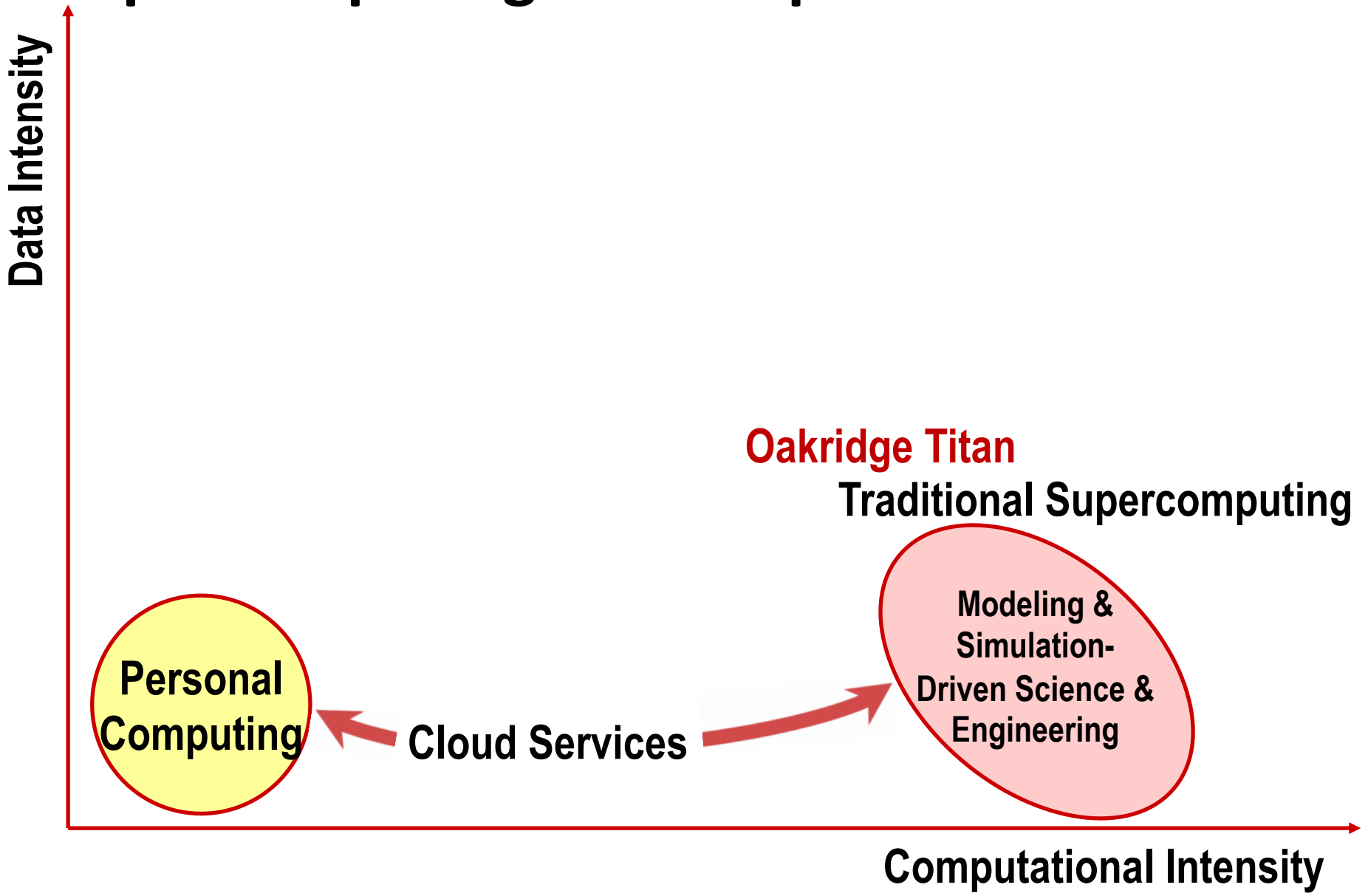
Sunway Taihu Light

- Chinese supercomputer, ranked 15th in the TOP500 list

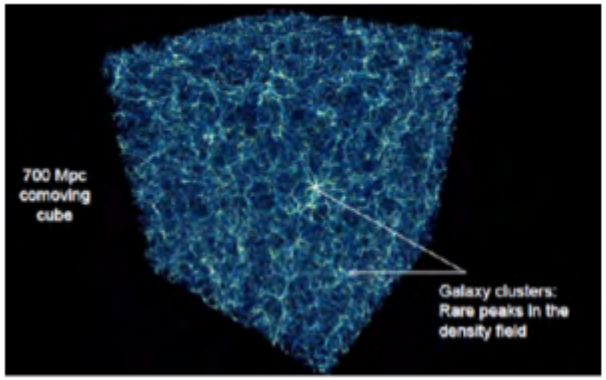
Computing Landscape



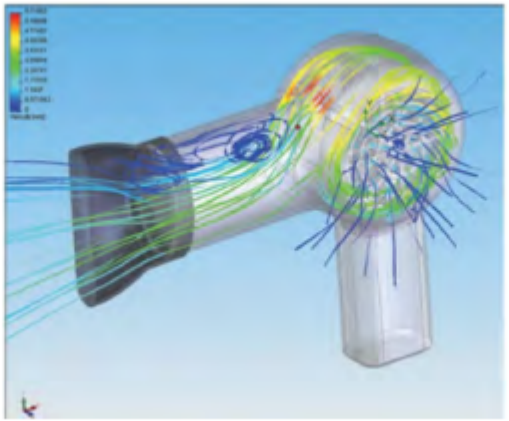
Supercomputing Landscape



Supercomputer Applications



Science

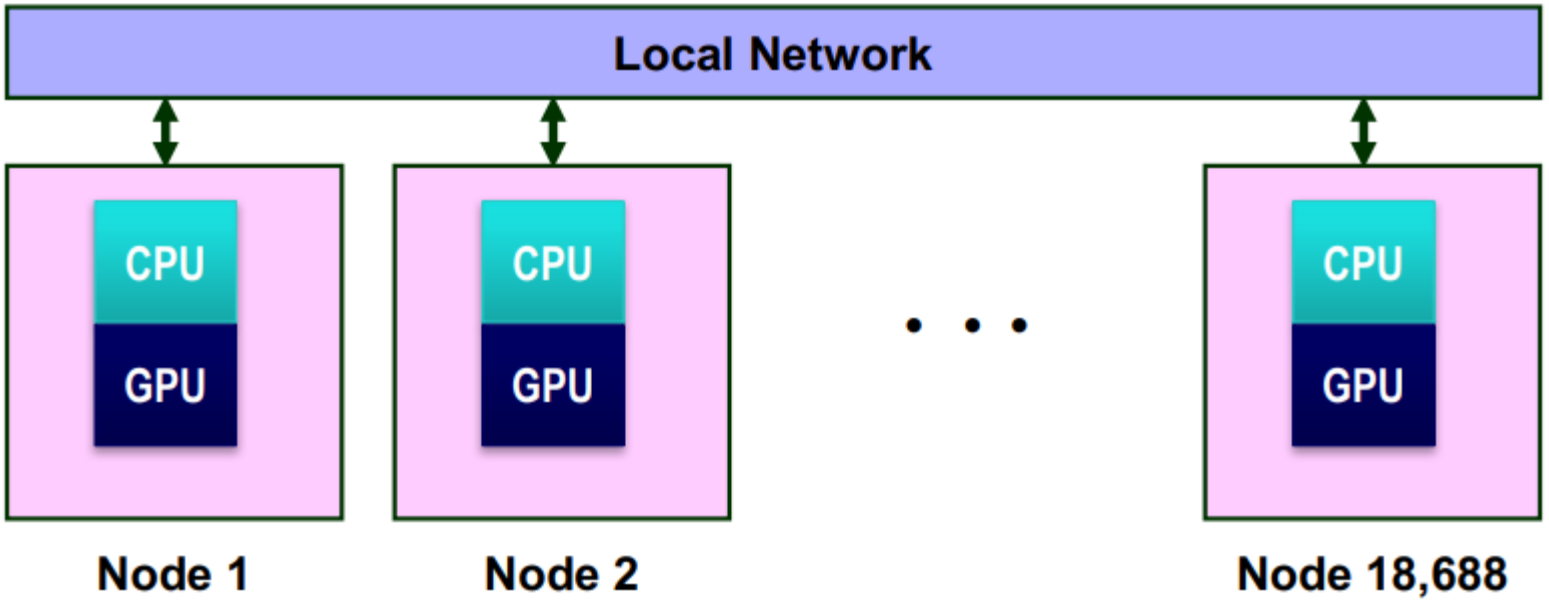


Industrial Products

Public Health

- **Simulation-Based Modeling**
  - System structure + initial conditions + transition behavior § Discretize time and space § Run simulation to see what happens
- **Requirements**
  - Model accurately reflects actual system § Simulation faithfully captures model

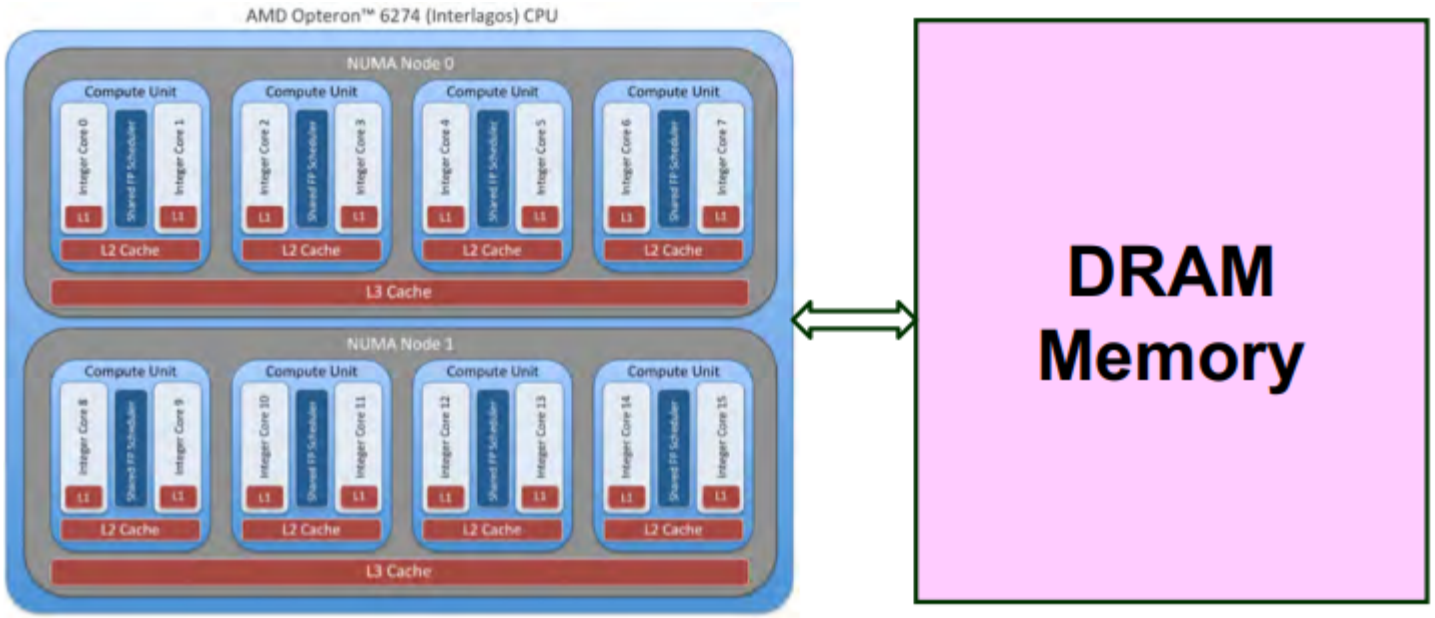
Titan Hardware



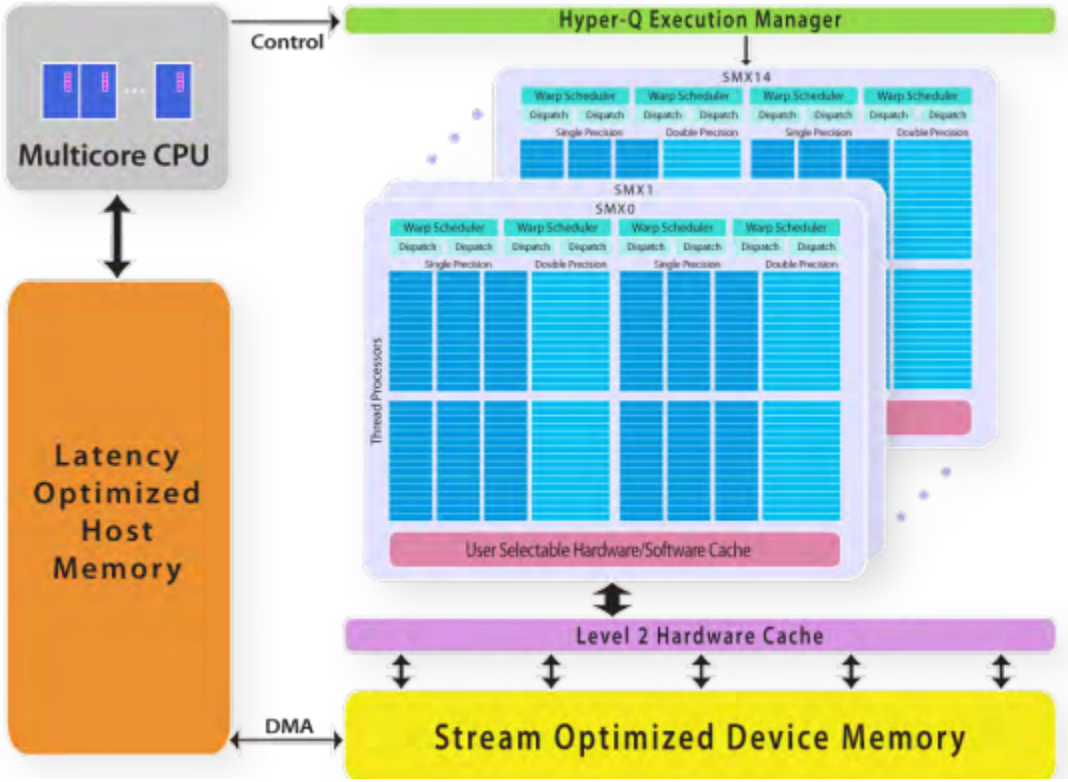
- **Each Node**
  - AMD 16-core processor
  - nVidia Graphics Processing Unit
  - 38 GB DRAM
  - No disk drive
- **Overall**
  - 7MW, $200M

Titan Node Structure: CPU

- **CPU**
  - 16 cores sharing common memory
  - Supports multithreaded programming
  - ~0.16 x 10^{12} floating-point operations per second (FLOPS) peak performance



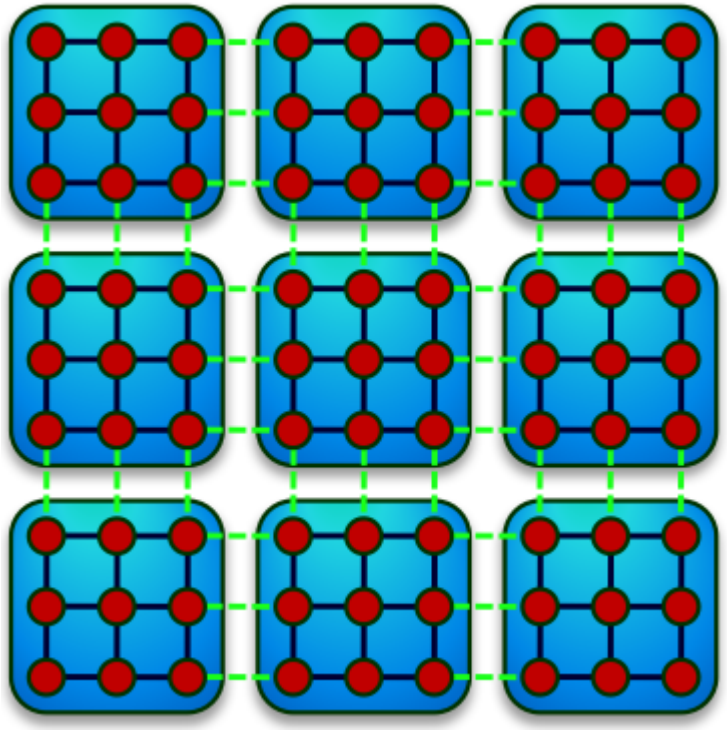
Titan Node Structure: GPU



- **Kepler GPU**
  - 14 multiprocessors
  - Each with 12 groups of 16 stream processors
    - 14 X 12 X 16 = 2688
  - Single-Instruction, Multiple-Data parallelism
    - Single instruction controls all processors in group
  - 4.0 x 1012 FLOPS peak performance

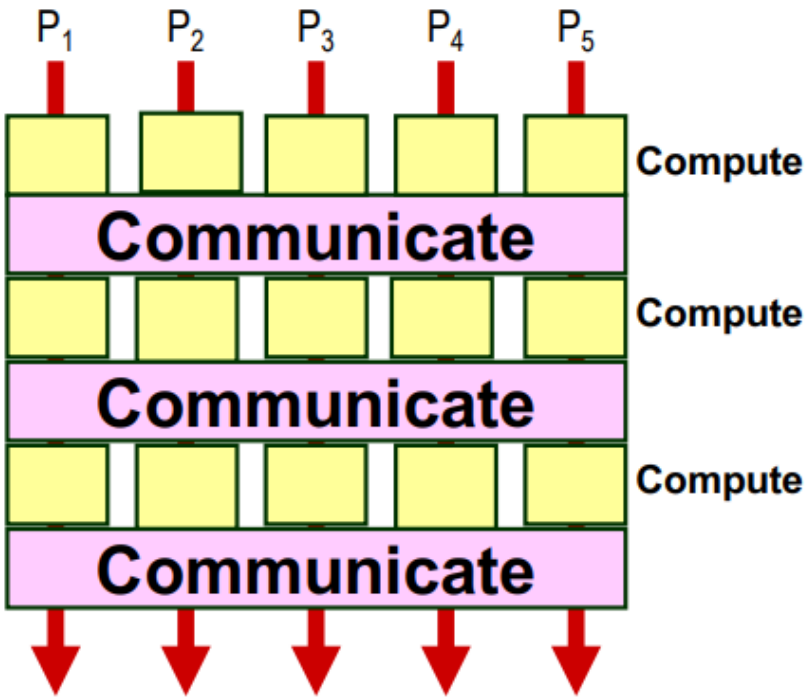
Titan Programming: Principle

- **Solving Problem Over Grid**
  - E.g., finite-element system
  - Simulate operation over time
- **Bulk Synchronous Model**
  - Partition into Regions
    - p regions for p-node machine
  - Map Region per Processor



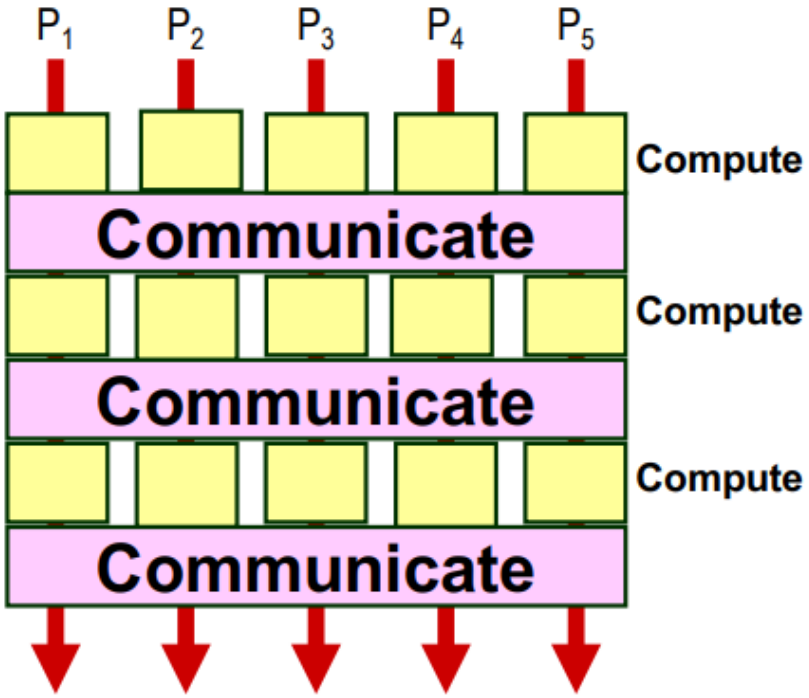
Titan Programming: Principle (cont)

- **Bulk Synchronous Model**
  - Map Region per Processor
  - Alternate
    - All nodes compute behavior of region
      - Perform on GPUs
    - All nodes communicate values at boundaries



Bulk Synchronous Performance

- Limited by performance of slowest processor
- **Strive to keep perfectly balanced**
  - Engineer hardware to be highly reliable
  - Tune software to make as regular as possible
  - Eliminate “noise”
    - Operating system events
    - Extraneous network activity



General Architecture of the Sunway TaihuLight

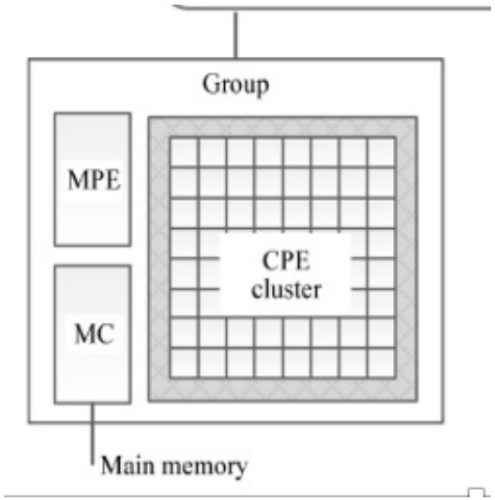
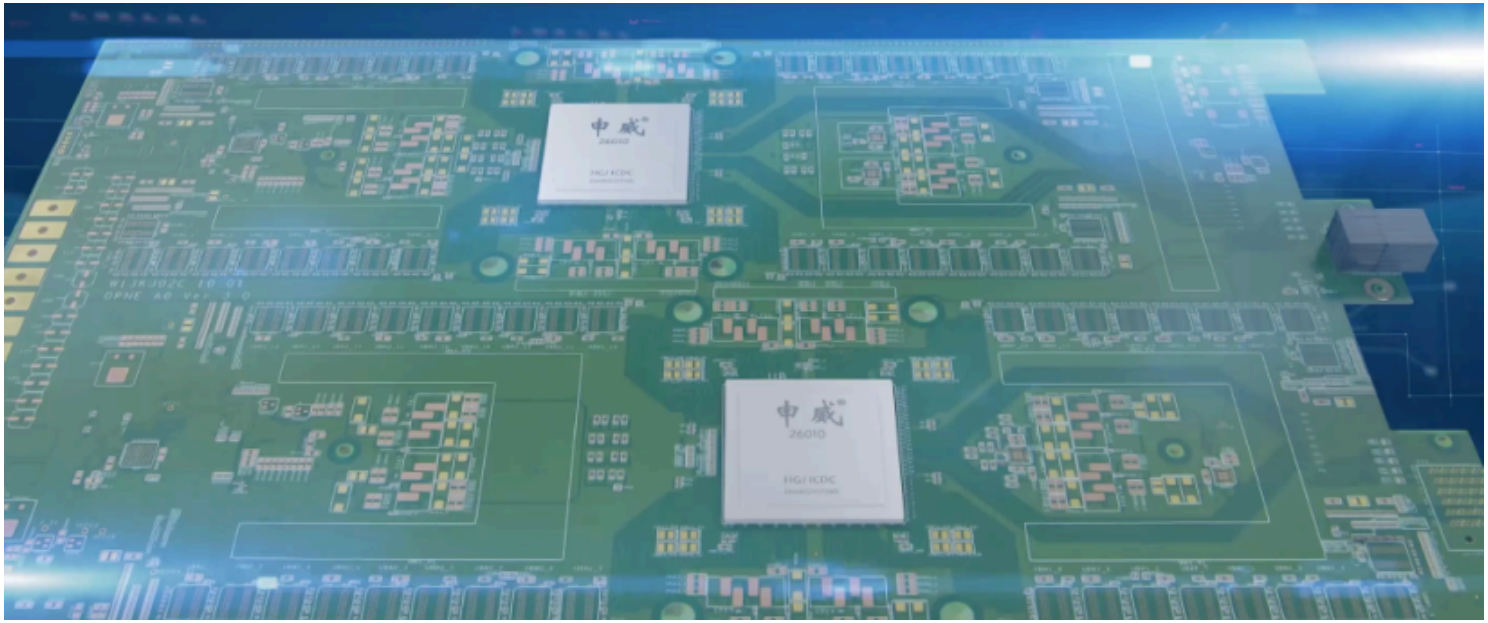
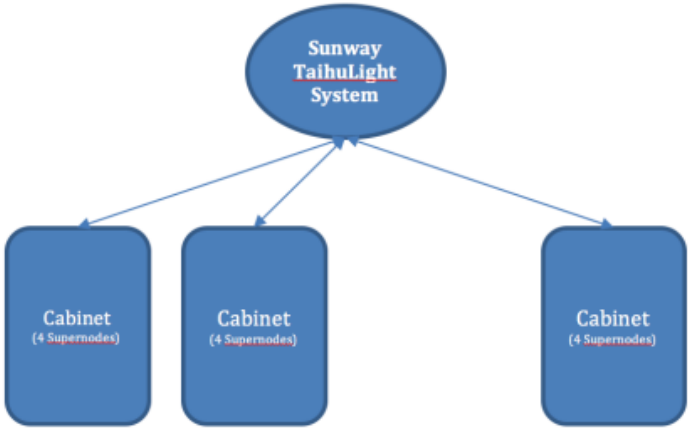
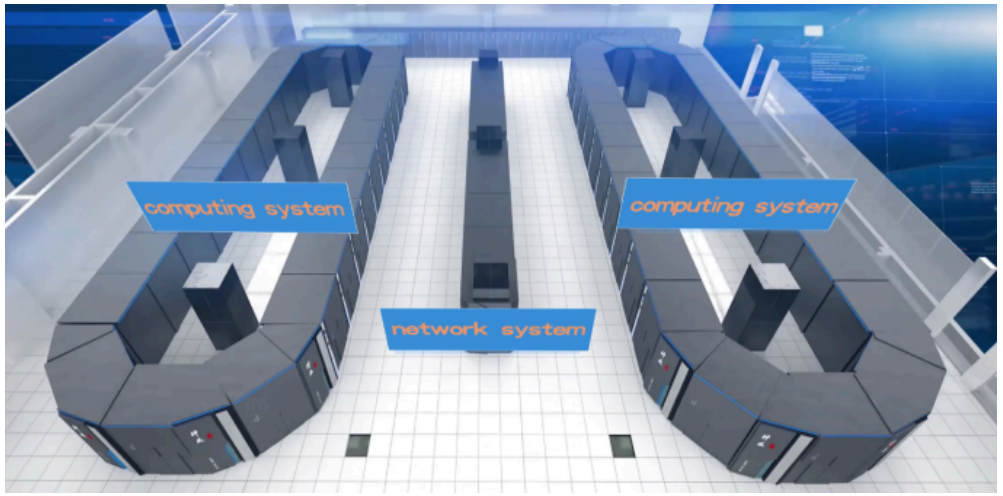


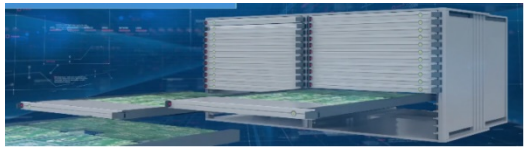
Figure 1: Core Group for Node

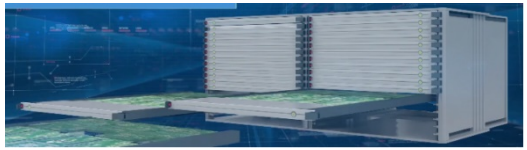


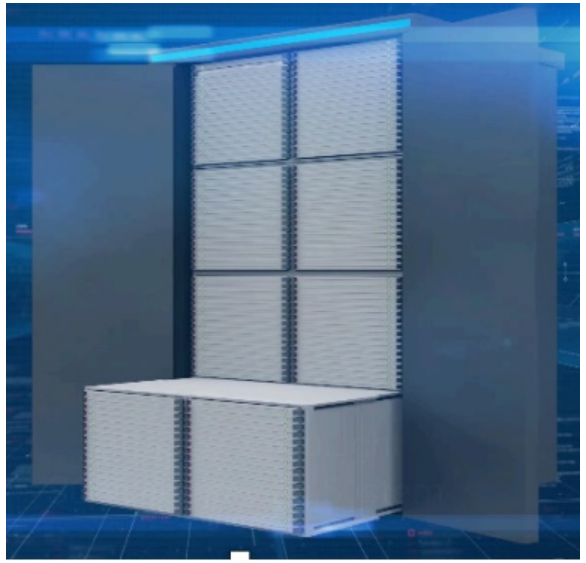
General Architecture of the Sunway TaihuLight



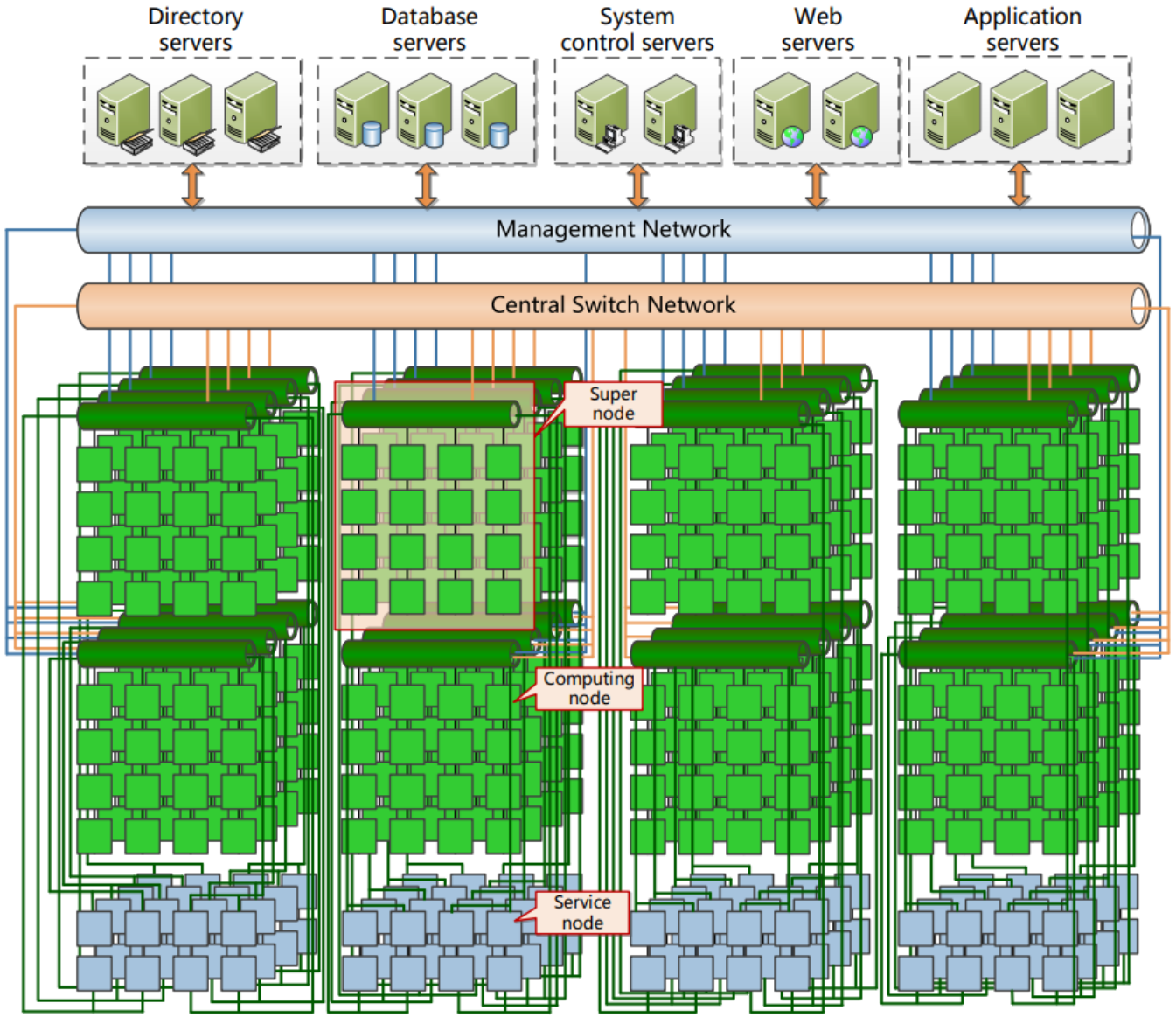




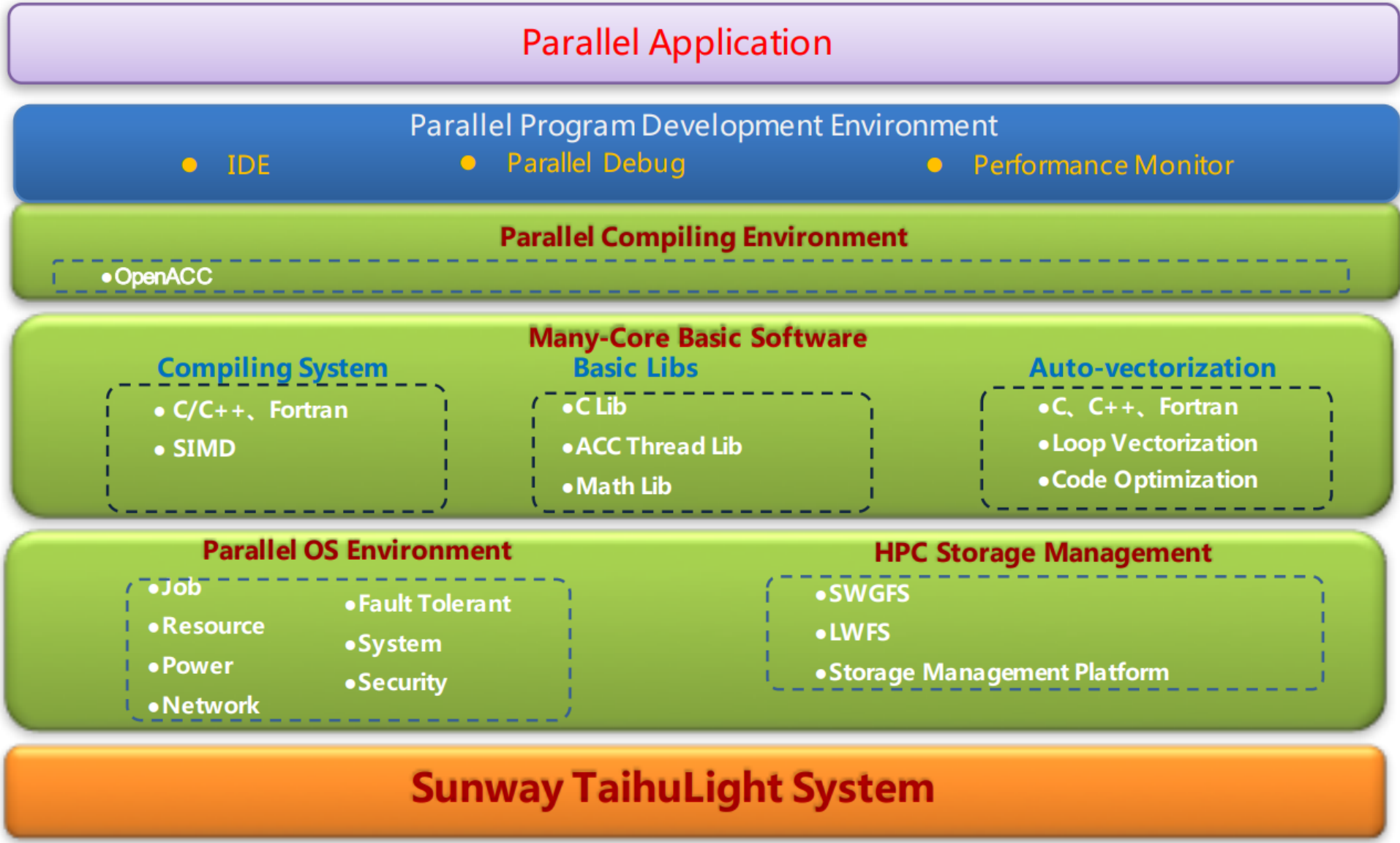




General Architecture of the Sunway TaihuLight

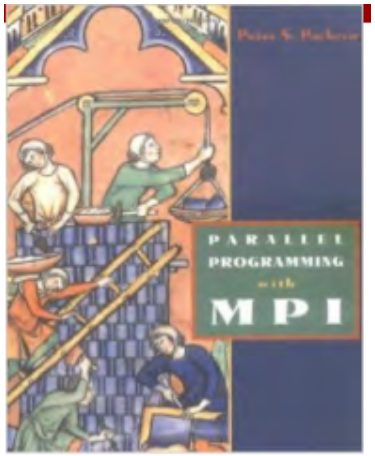


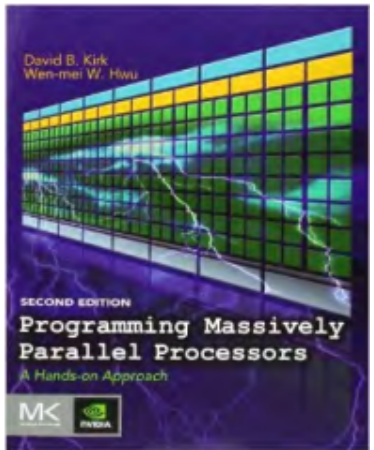
Software Stack of the Sunway TaihuLight



Supercomputer Programming

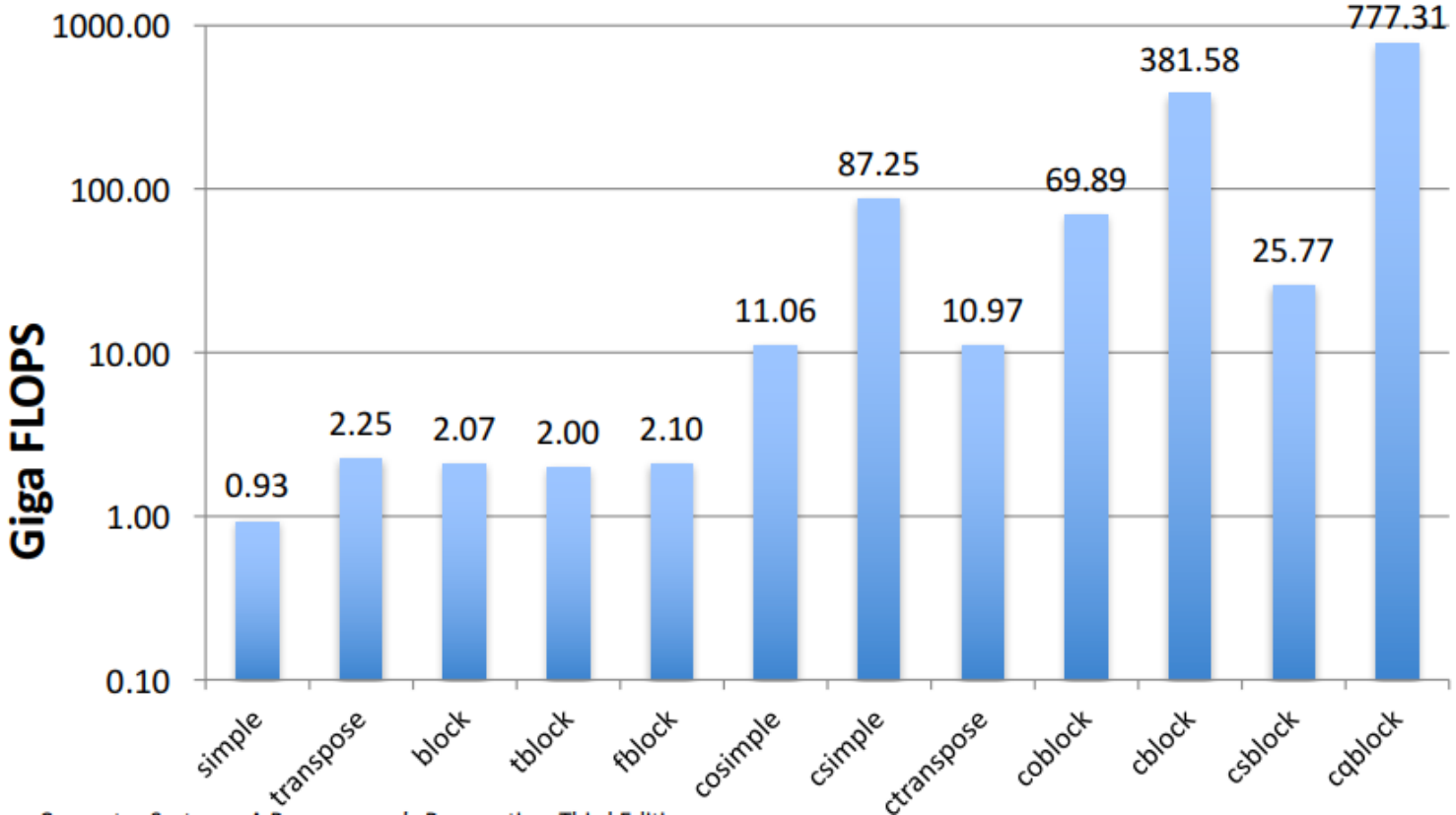
- **System Level**
  - Message-Passing Interface (MPI) supports node computation, synchronization and communication
- **Node Level**
  - OpenMP supports thread-level operation of node CPU
  - CUDA programming environment for GPUs
    - Performance degrades quickly if don't have perfect balance among memories and processors
- **Result**
  - Single program is complex combination of multiple programming paradigms
  - Tend to optimize for specific hardware configuration





GPU Experience on Matrix Multiply

- **Multiply two 1024 x 1024 matrices (MM)**
  - 2 X 10^9 floating point operations
  - Express performance in Giga FLOPS
  - Program in CUDA and map onto nVidia GPU

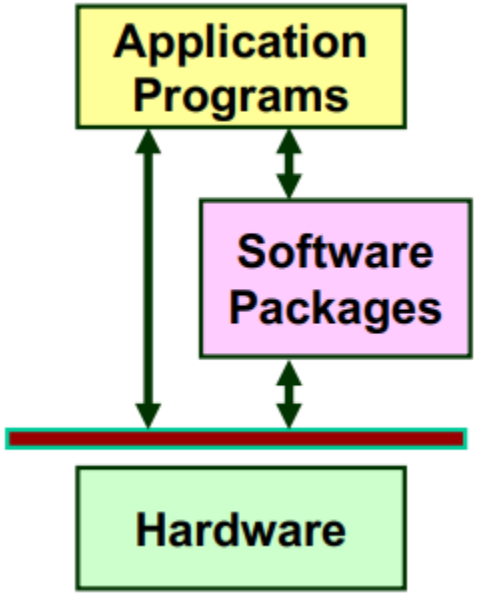


Matrix Multiplication Progress

- **Versions**
  - Naive 1
  - Simple parallel 11
  - Blocking 70
  - nVidia Example Code 388
  - Reorient memory accesses 382
  - Packed data access 777
- **Observations**
  - Progress is very nonlinear
    - Not even monotonic
  - Requires increased understanding of how program maps onto hardware
  - Becomes more specialized to specific hardware configuration

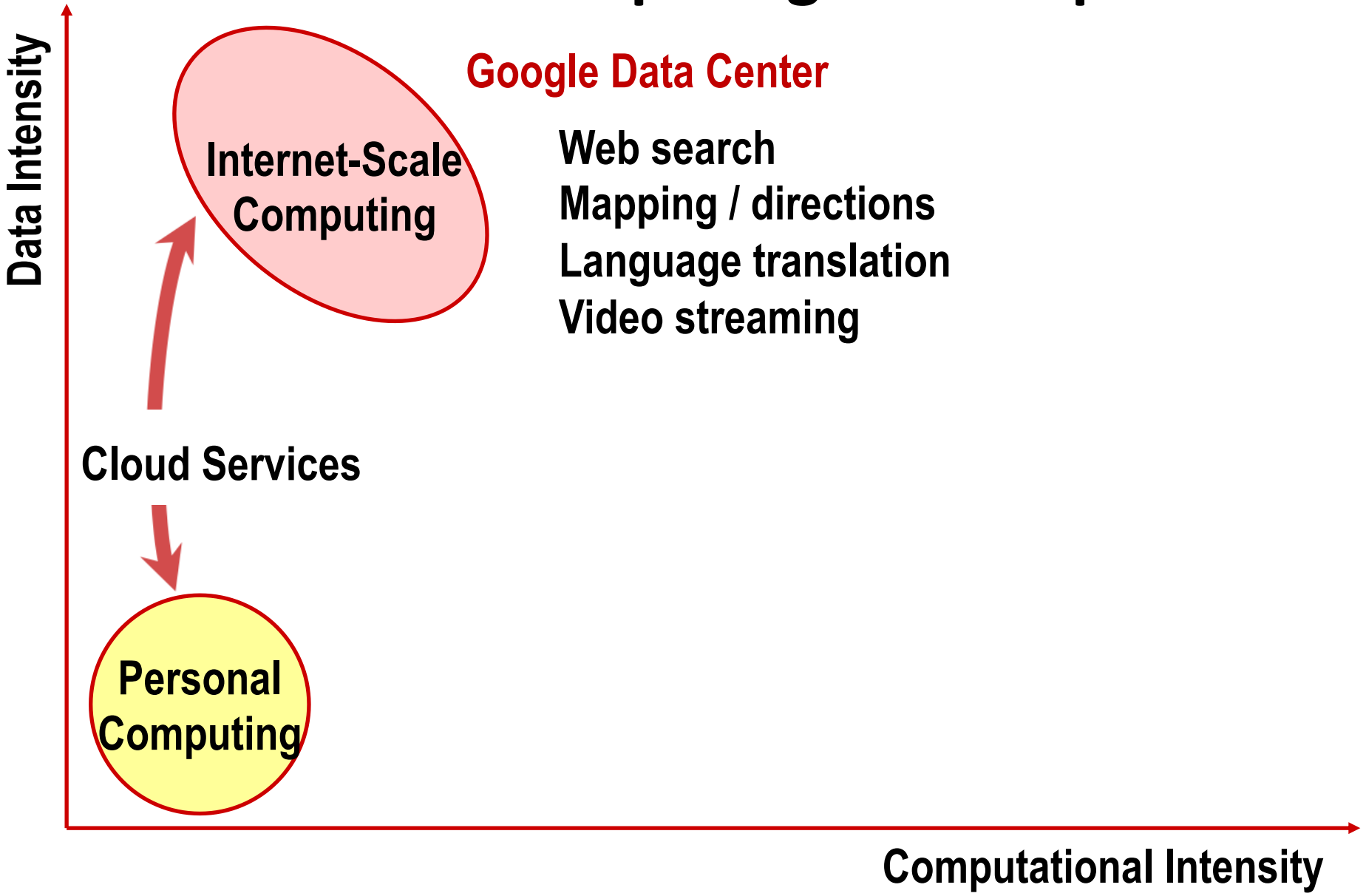
Supercomputer Programming Model

- **Program on top of bare hardware**
- **Performance**
  - Low-level programming to maximize node performance
  - Keep everything globally synchronized and balanced
- **Reliability**
  - Single failure causes major delay
  - Engineer hardware to minimize failures



Machine-Dependent Programming Mode

Data-Intensive Computing Landscape



Internet Computing

- **Web Search**
  - Aggregate text data from across WWW
  - No definition of correct operation
  - Do not need real-time updating

- **Mapping Services**
  - Huge amount of (relatively) static data
  - Each customer requires individualized computation

- **Online Documents**
  - Must be stored reliably
  - Must support real-time updating
  - (Relatively) small data volumes

Other Data-Intensive Computing Applications

- **Wal-Mart**
  - 267 million items/day, sold at 6,000 stores
  - HP built them 4 PB data warehouse
  - Mine data to manage supply chain, understand market trends, formulate pricing strategies

- **LSST**
  - Chilean telescope will scan entire sky every 3 days
  - A 3.2 gigapixel digital camera
  - Generate 30 TB/day of image data

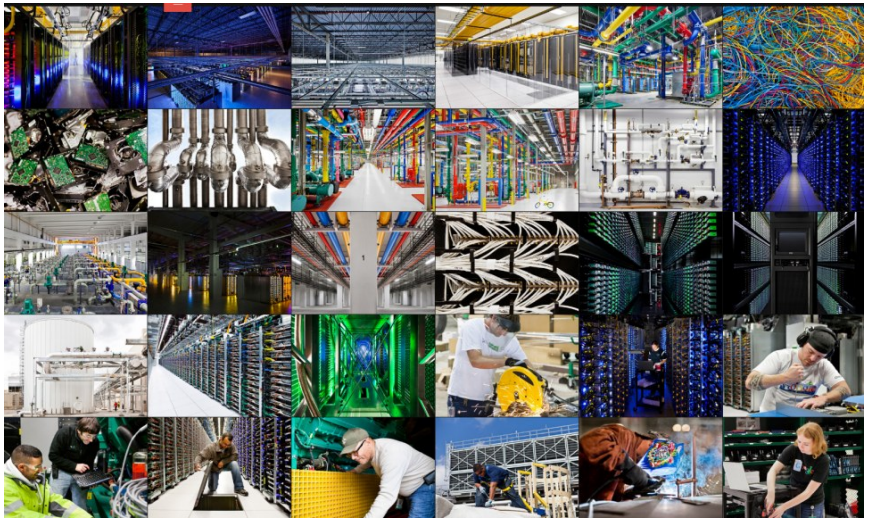
LSST
Legacy Survey of Space and Time

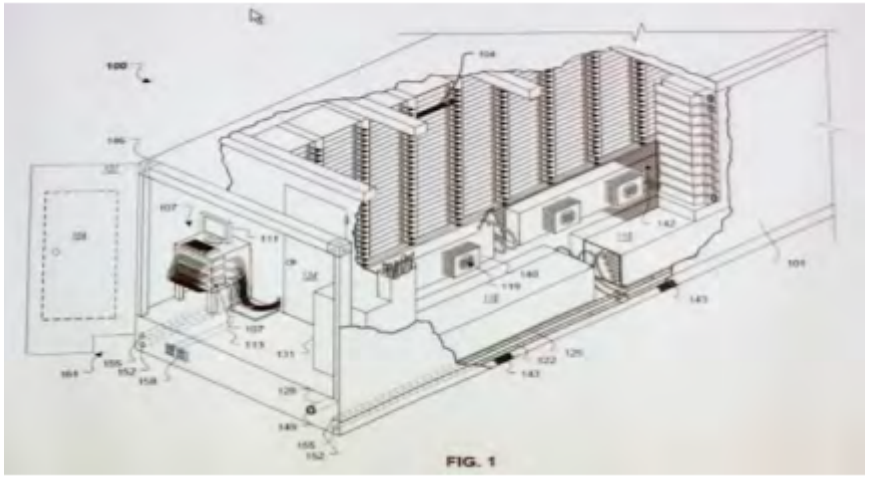
Data-Intensive Application Characteristics

- **Diverse Classes of Data**
  - Structured & unstructured
  - High & low integrity requirements
- **Diverse Computing Needs**
  - Localized & global processing
  - Numerical & non-numerical
  - Real-time & batch processing

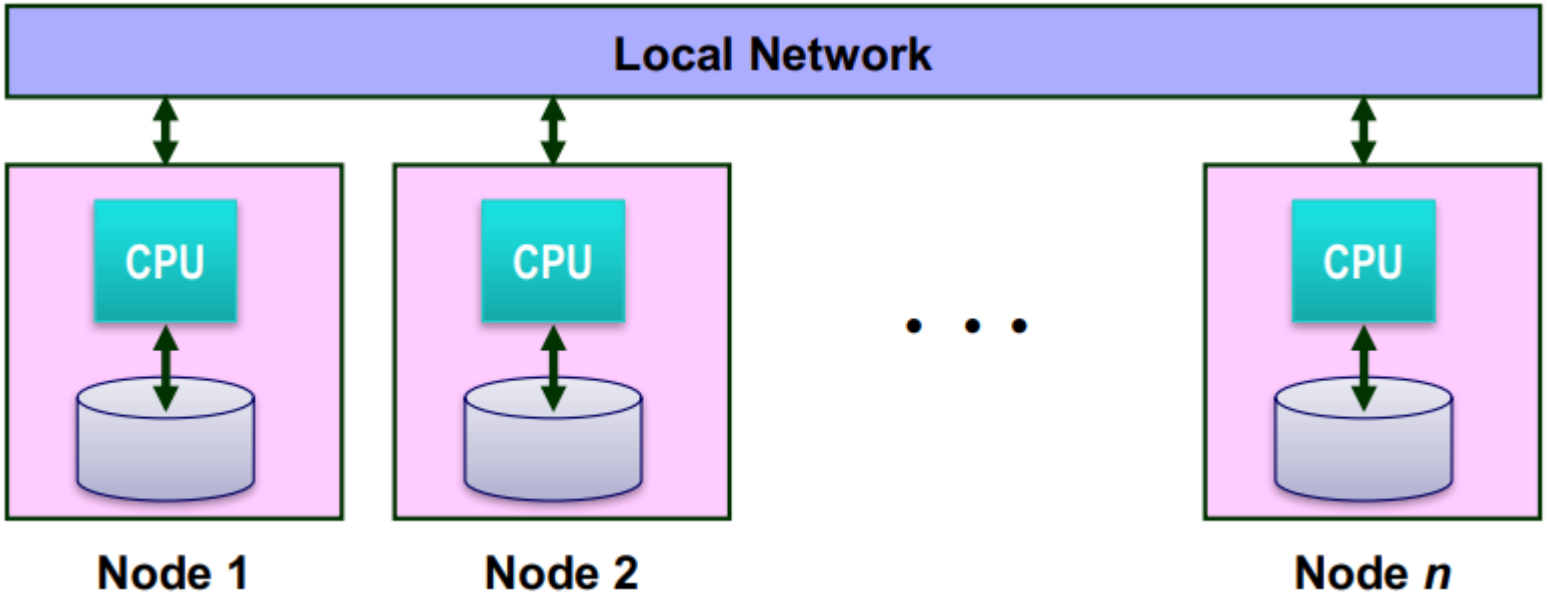
Google Data Centers

- **Dalles, Oregon**
  - Hydroelectric power @ 2¢ / KW Hr
  - 50 Megawatts
  - Enough to power 60,000 homes
  - Engineered for low cost, modularity & power efficiency
  - Container: 1160 server nodes, 250KW



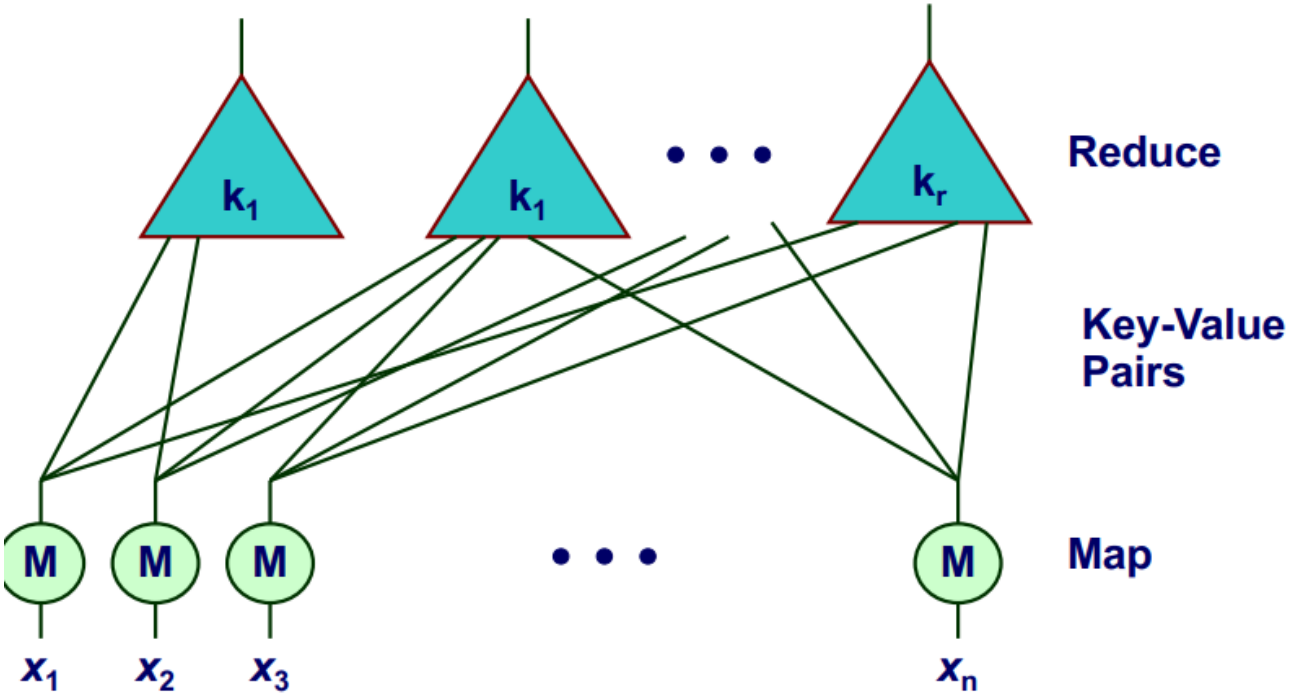


Google Cluster



- **Typically 1,000–2,000 nodes**
- **Node Contains**
  - 2 multicore CPUs
  - 2 disk drives
  - DRAM

Map/Reduce Programming Model



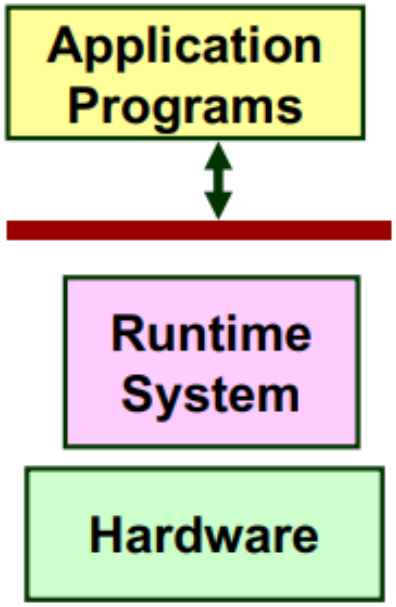
- **Map computation across many objects**
  - E.g., 10^{10} Internet web pages
- **Aggregate results in many different ways**
- **System deals with issues of resource allocation & reliability**

Dean & Ghemawat: "MapReduce: Simplified Data Processing on Large Clusters", OSDI 2004

Cluster Programming Model

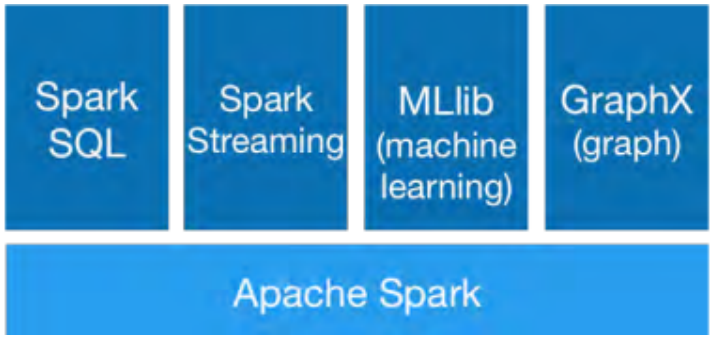
- Application programs written in terms of high-level operations on data
- Runtime system controls scheduling, load balancing, …

Machine-Independent Programming Model



- **Scaling Challenges**
  - Centralized scheduler forms bottleneck
  - Copying to/from disk very costly
  - Hard to limit data movement
    - Significant performance factor

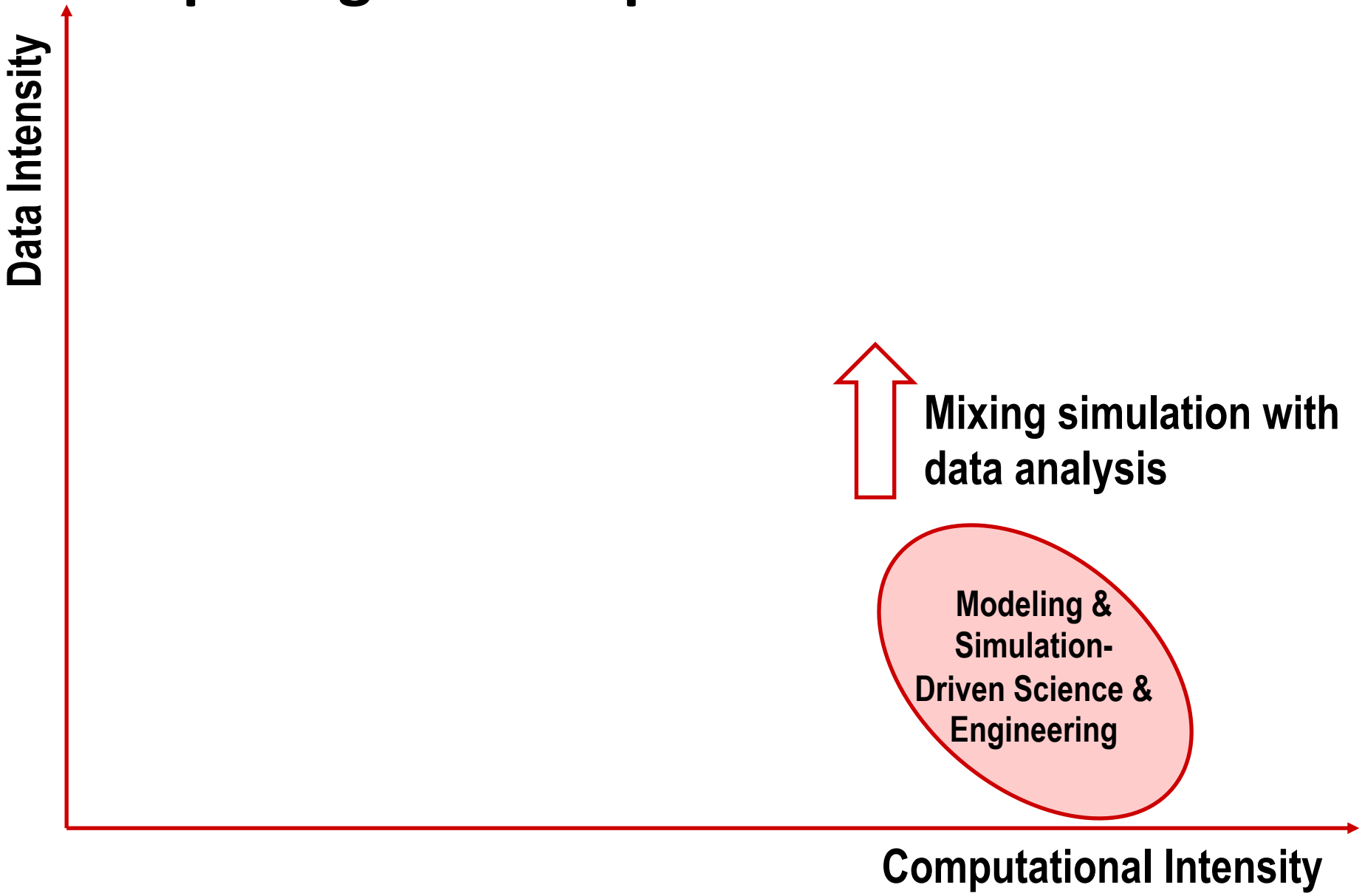
Some Typical Programming Systems



- Spark Project
- at U.C., Berkeley
- Grown to have large open source community

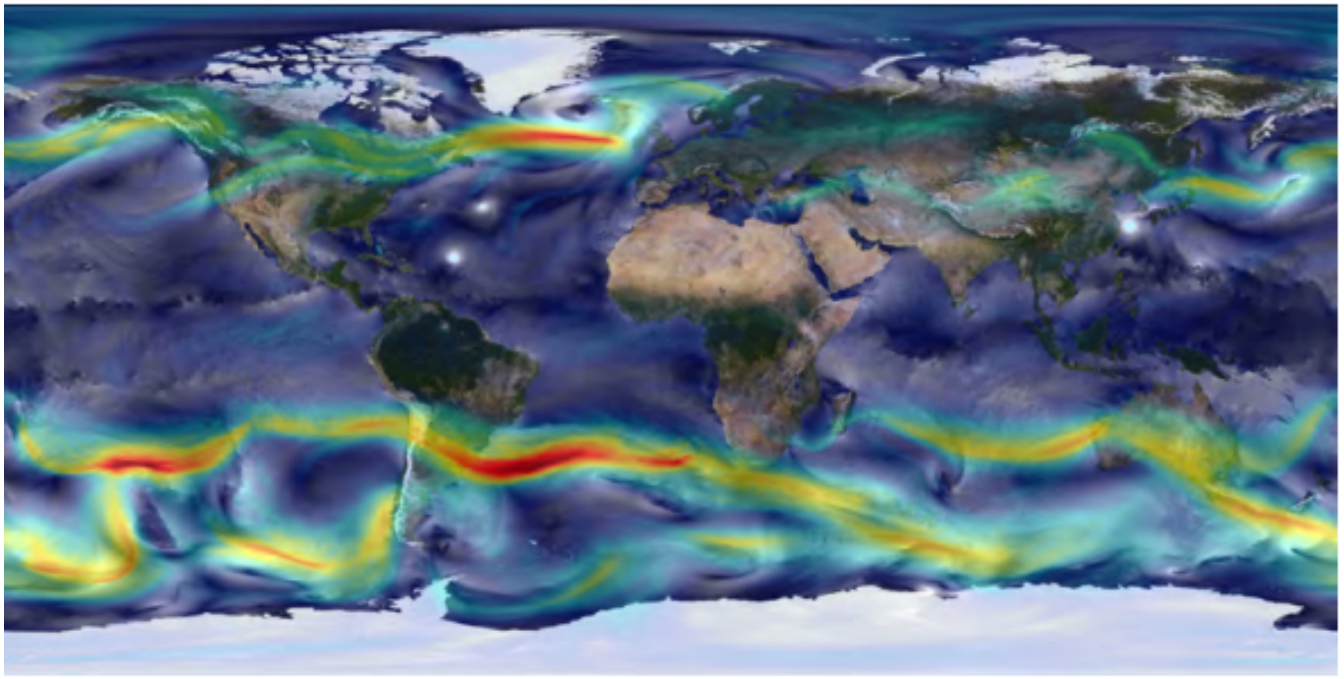
- GraphLab
- Started as project at CMU by Carlos Guestrin
- Environment for describing machine-learning algorithms
  - Sparse matrix structure described by graph
  - Computation based on updating of node values

Computing Landscape Trends

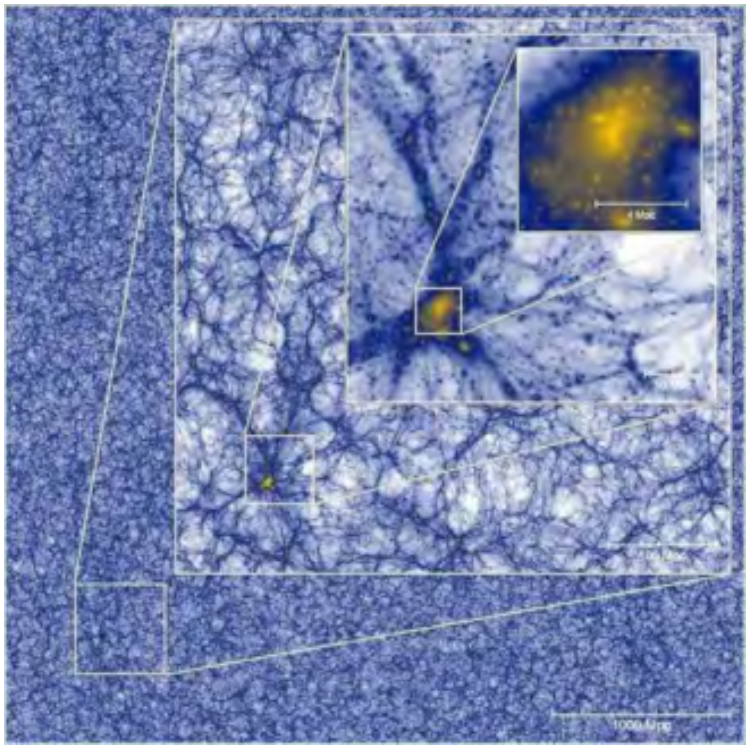


Combining Simulation with Real Data

- **Limitations**
  - Simulation alone: Hard to know if model is correct
  - Data alone: Hard to understand causality & “what if”
- **Combination**
  - Check and adjust model during simulation

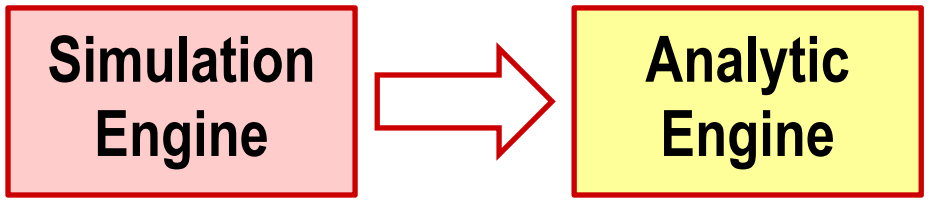


Real-Time Analytics

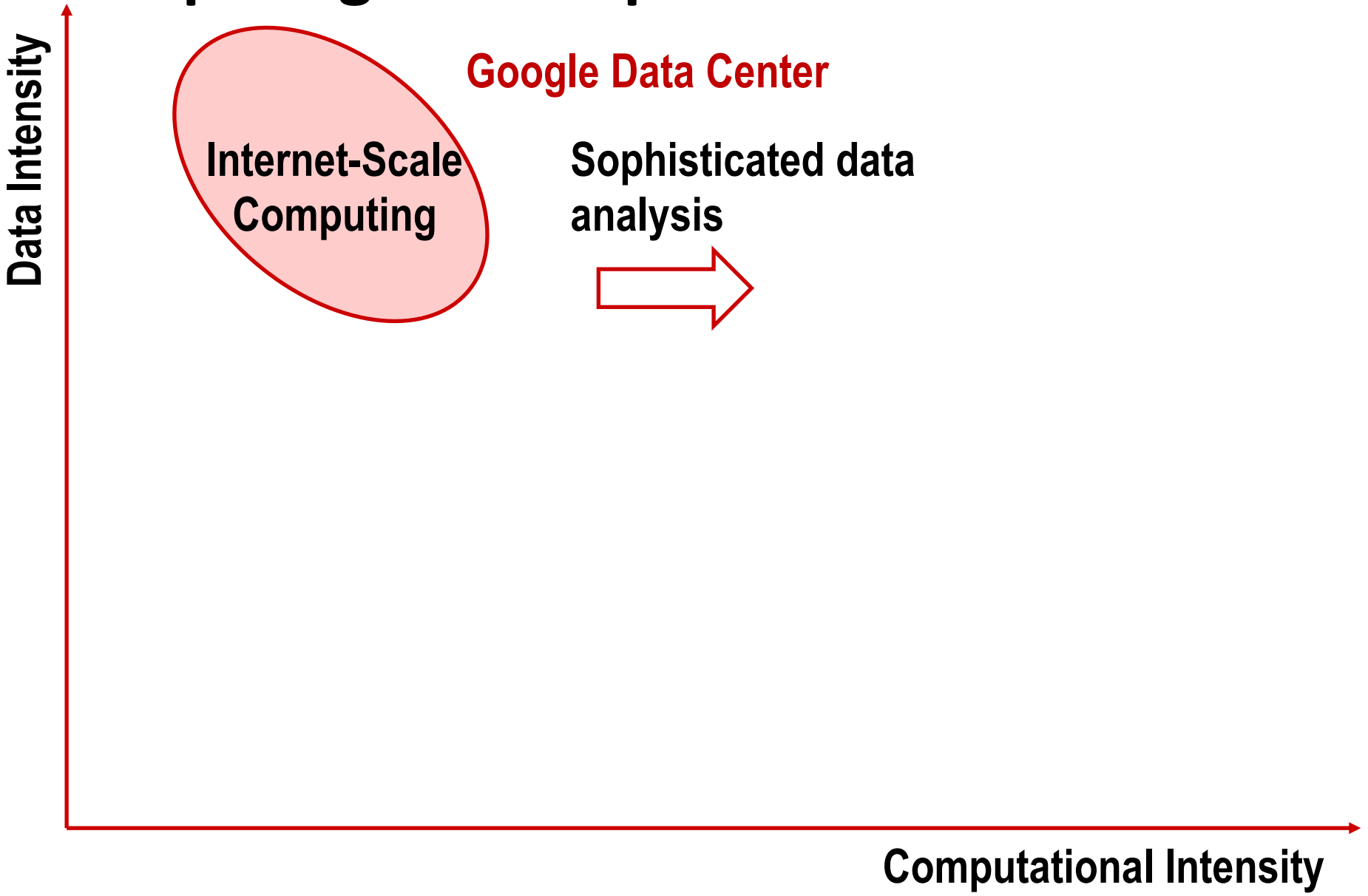


- **Millenium XXL Simulation (2010)**
  - 3 X 10^9 particles
  - Simulation run of 9.3 days on 12,228 cores
  - 700TB total data generated
  - Save at only 4 time points
  - 70 TB
  - Large-scale simulations generate large data sets

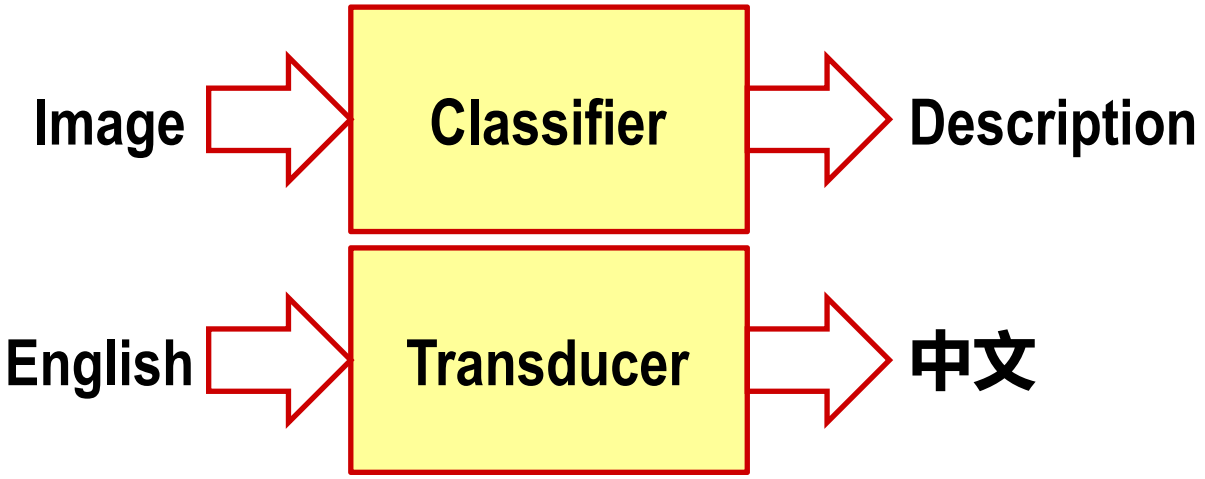
- **What If?**
  - Could perform data analysis while simulation is running

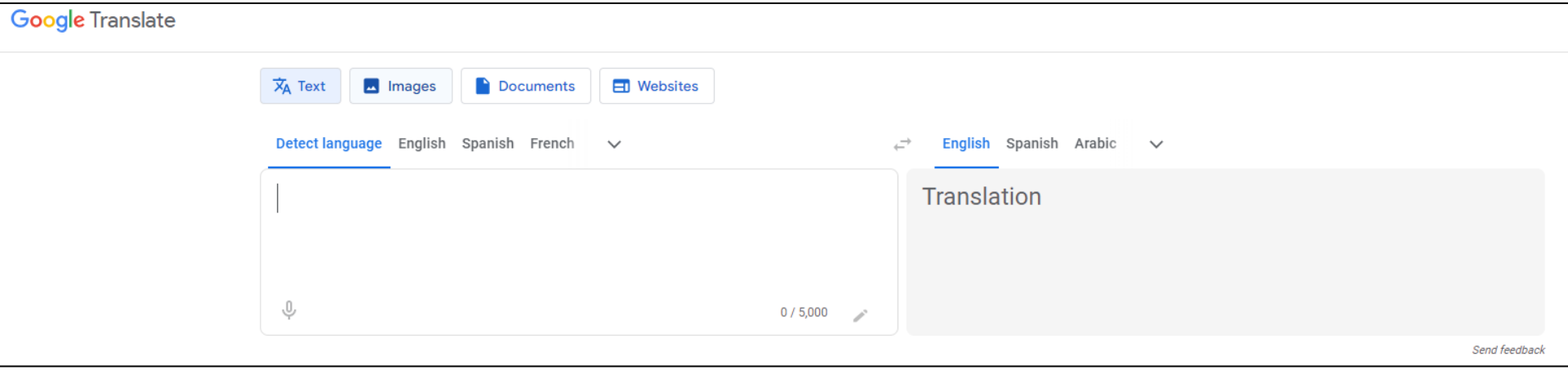


Computing Landscape Trends



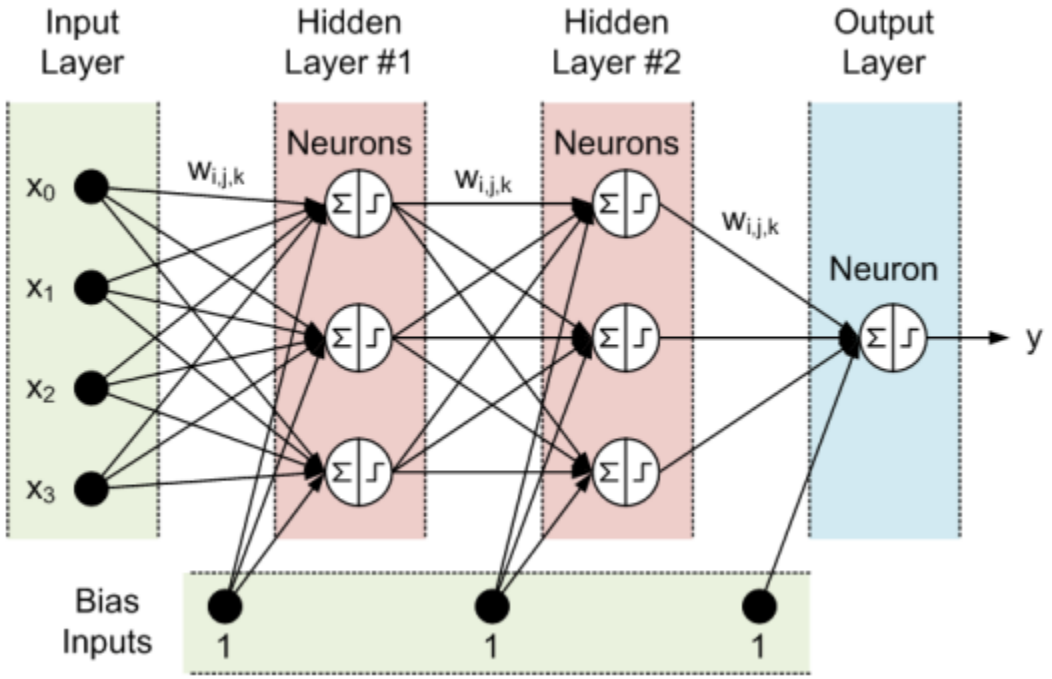
Example Analytic Applications





Data Analysis with Deep Neural Networks

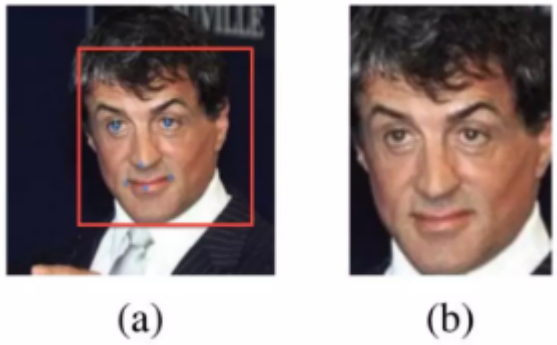
- **Task:**
  - Compute classification of set of input signals

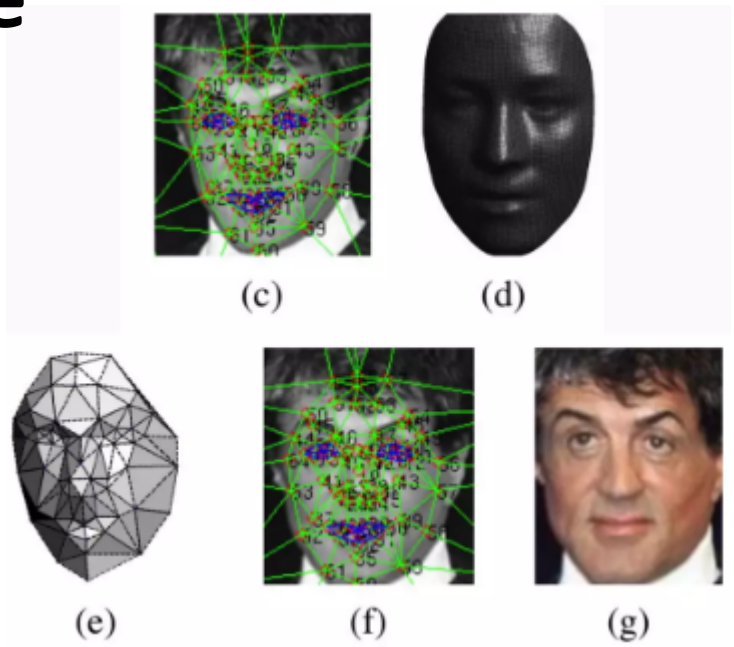


- **Training**
  - Use many training samples of form input / desired output
  - Compute weights that minimize classification error
- **Operation**
  - Propagate signals from input to output

DNN Application Example

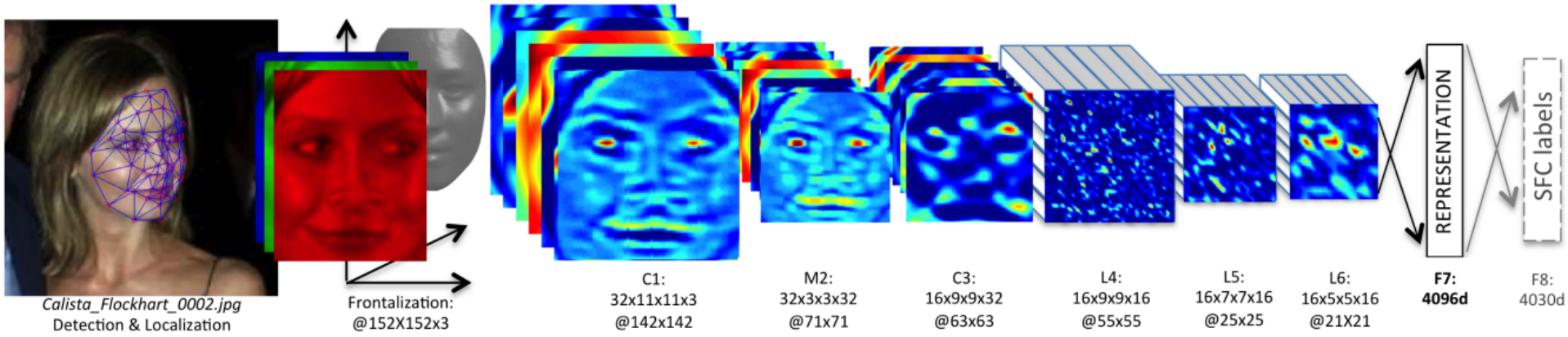
- Facebook DeepFace Architecture





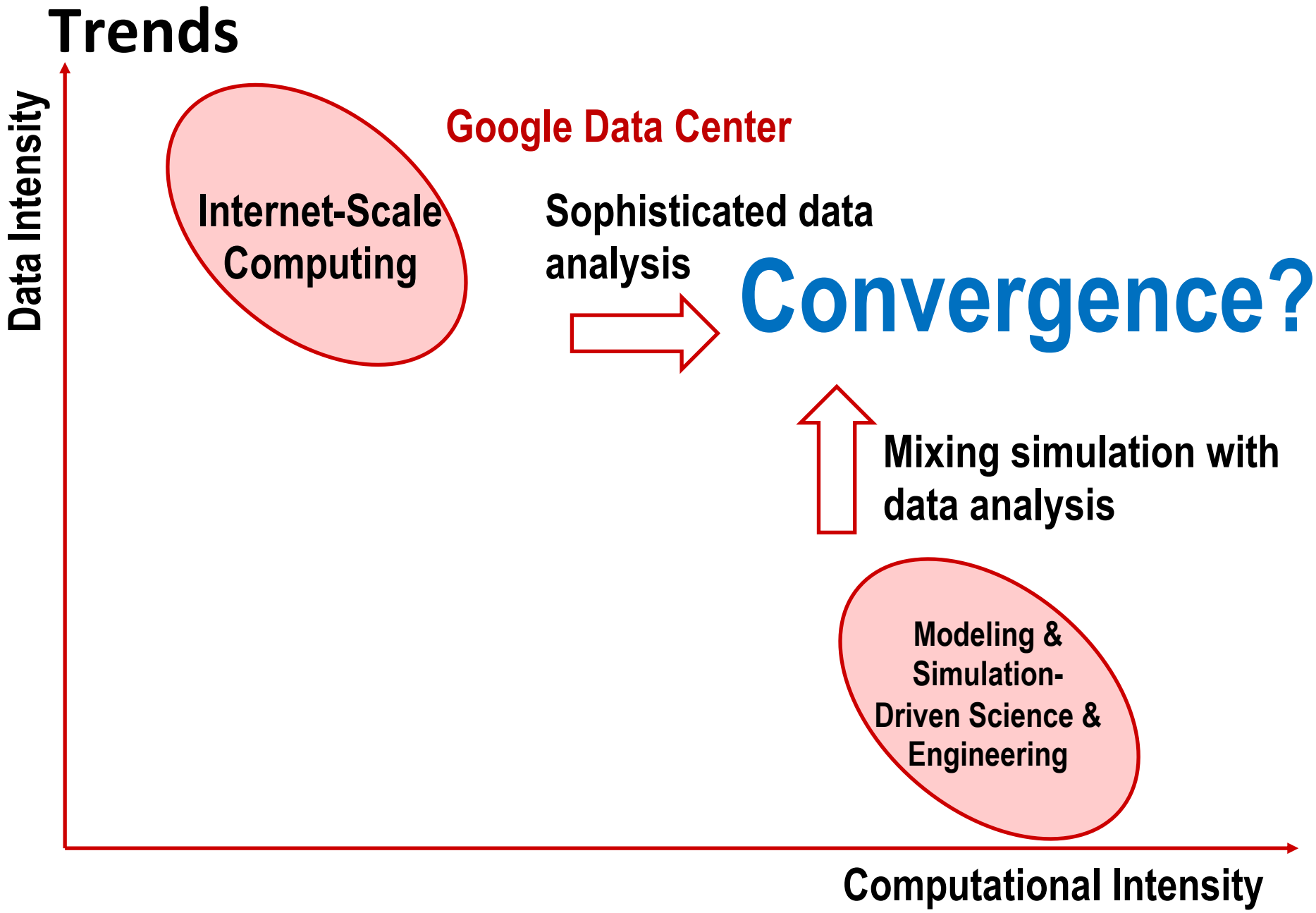
2D-alignment

3D-alignment



Training DNNs

- **Characteristics**
  - Iterative numerical algorithm
  - Regular data organization
- **Project Adam Training**
  - 2B connections
  - 15M images
  - 62 machines
  - 10 days


Trends
Data Intensity
Internet-Scale Computing
Google Data Center
Sophisticated data analysis
Convergence?
Mixing simulation with data analysis
Modeling & Simulation-Driven Science & Engineering
Computational Intensity

Challenges for Convergence

Supercomputers	Data Center Clusters
Hardware	
■ Customized	■ Consumer grade
■ Optimized for reliability	■ Optimized for low cost
Run-Time System	
■ Source of “noise”	■ Provides reliability
■ Static scheduling	■ Dynamic allocation
Application Programming	
■ Low-level, processor-centric model	■ High level, data-centric model

Computation/Data Convergence

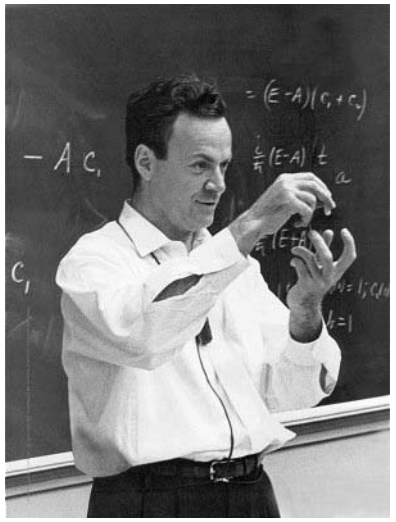
- **Two Important Classes of Large-Scale Computing**
  - Computationally intensive supercomputing
  - Data intensive processing
  - Internet companies + many other applications
- **Followed Different Evolutionary Paths**
  - Supercomputers: Get maximum performance from available hardware
  - Data center clusters: Maximize cost/performance over variety of datacentric tasks
  - Yielded different approaches to hardware, runtime systems, and application programming
- **A Convergence Would Have Important Benefits**
  - Computational and data-intensive applications
  - But, not clear how to do it

Future of Computing

QUANTUM COMPUTING

Introduction

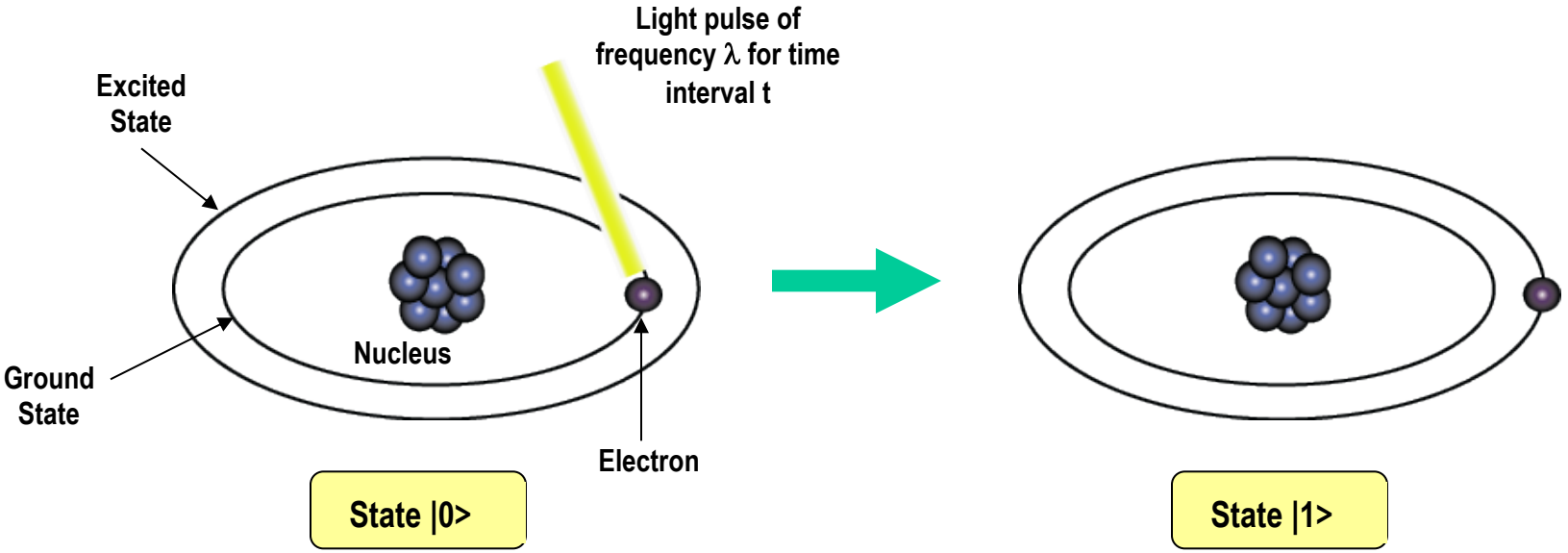
- "I think I can safely say that nobody understands quantum mechanics" - Feynman
- 1982 - Feynman proposed the idea of creating machines based on the laws of quantum mechanics instead of the laws of classical physics.
- 1985 - David Deutsch developed the quantum turing machine, showing that quantum circuits are universal.
- 1994 - Peter Shor came up with a quantum algorithm to factor very large numbers in polynomial time.
- 1997 - Lov Grover develops a quantum search algorithm with O(√N) complexity



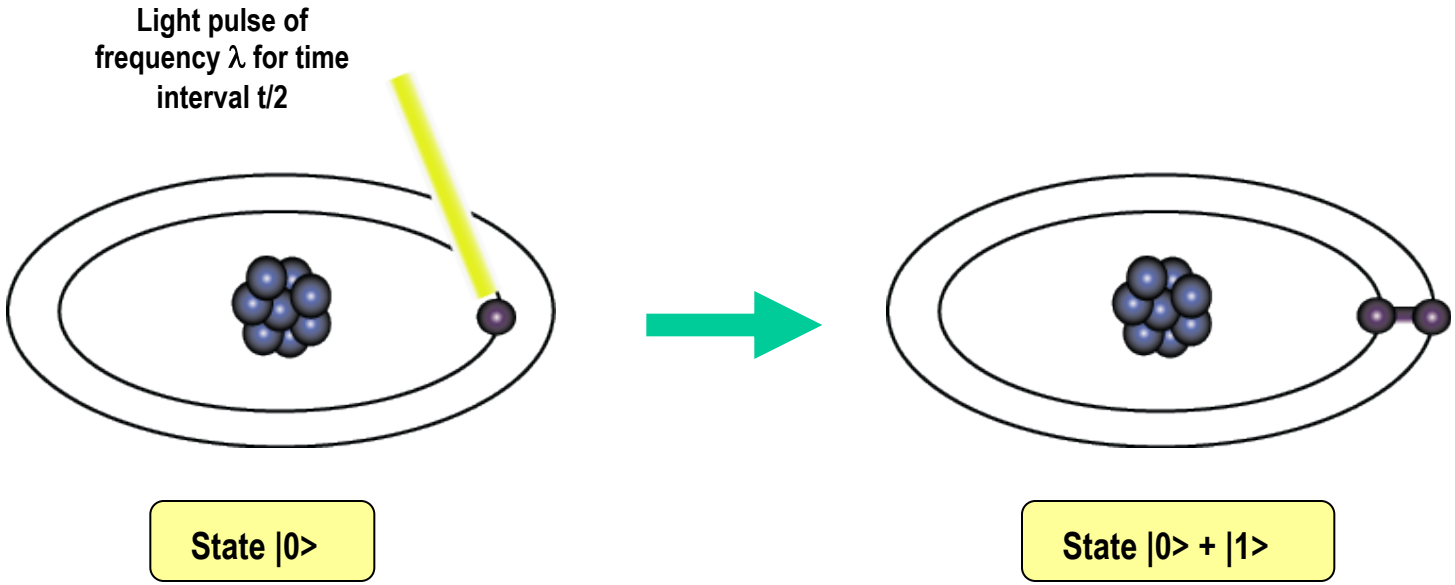
Representation of Data - Qubits

A bit of data is represented by a single atom that is in one of two states denoted by |0> and |1>. A single bit of this form is known as a *qubit*

A physical implementation of a qubit could use the two energy levels of an atom. An excited state representing |1> and a ground state representing |0>.



Representation of Data - Superposition

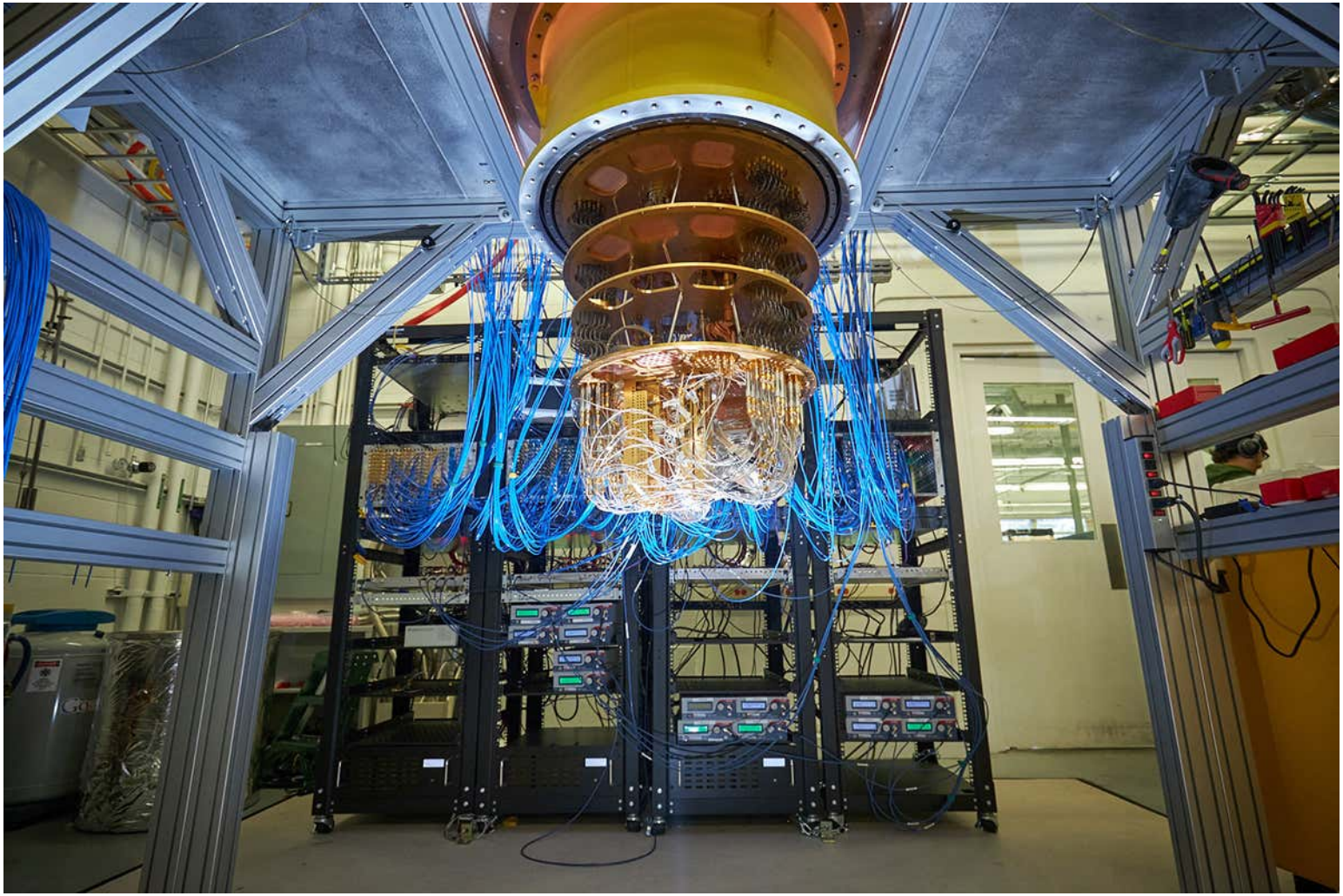


Consider a 3 bit qubit register. An equally weighted superposition of all possible states would be denoted by:

$$|\psi\rangle = \frac{1}{\sqrt{8}}|000\rangle + \frac{1}{\sqrt{8}}|001\rangle + \ldots + \frac{1}{\sqrt{8}}|111\rangle$$

Quantum Computing Machine

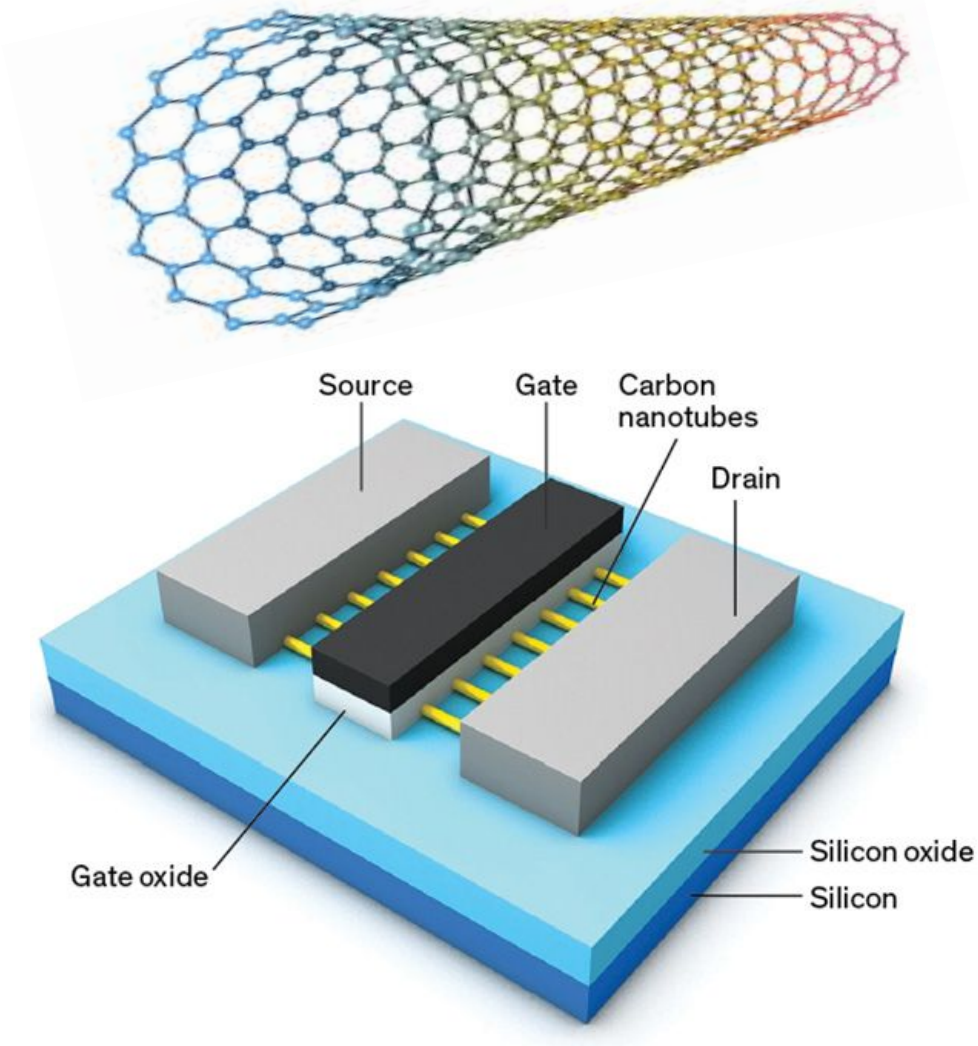
- Google's Sycamore quantum computer

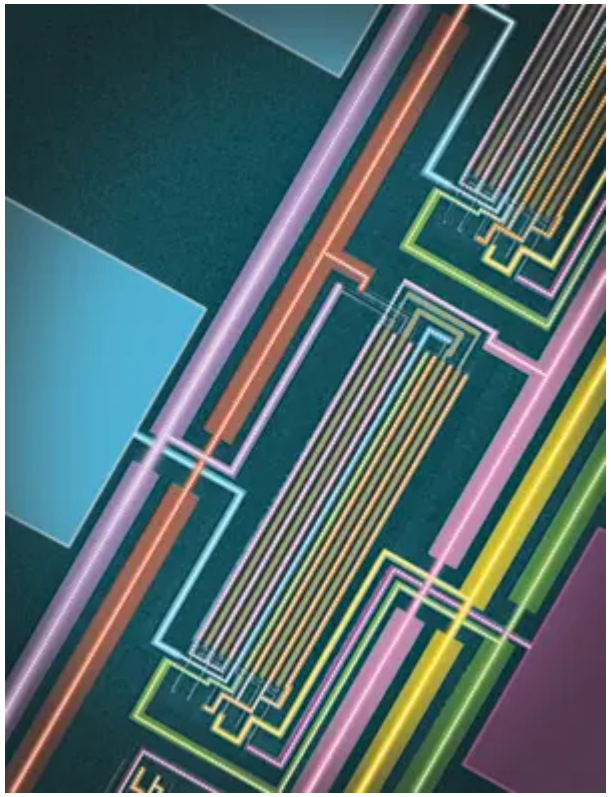


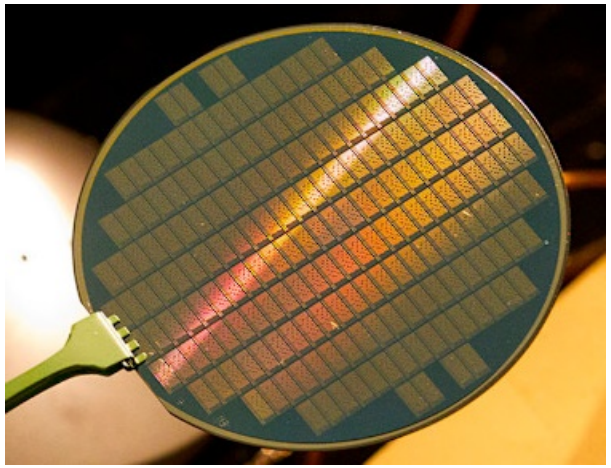
Future of Computing

CARBON NANOTUBE COMPUTER

CNT Transistor & Circuit

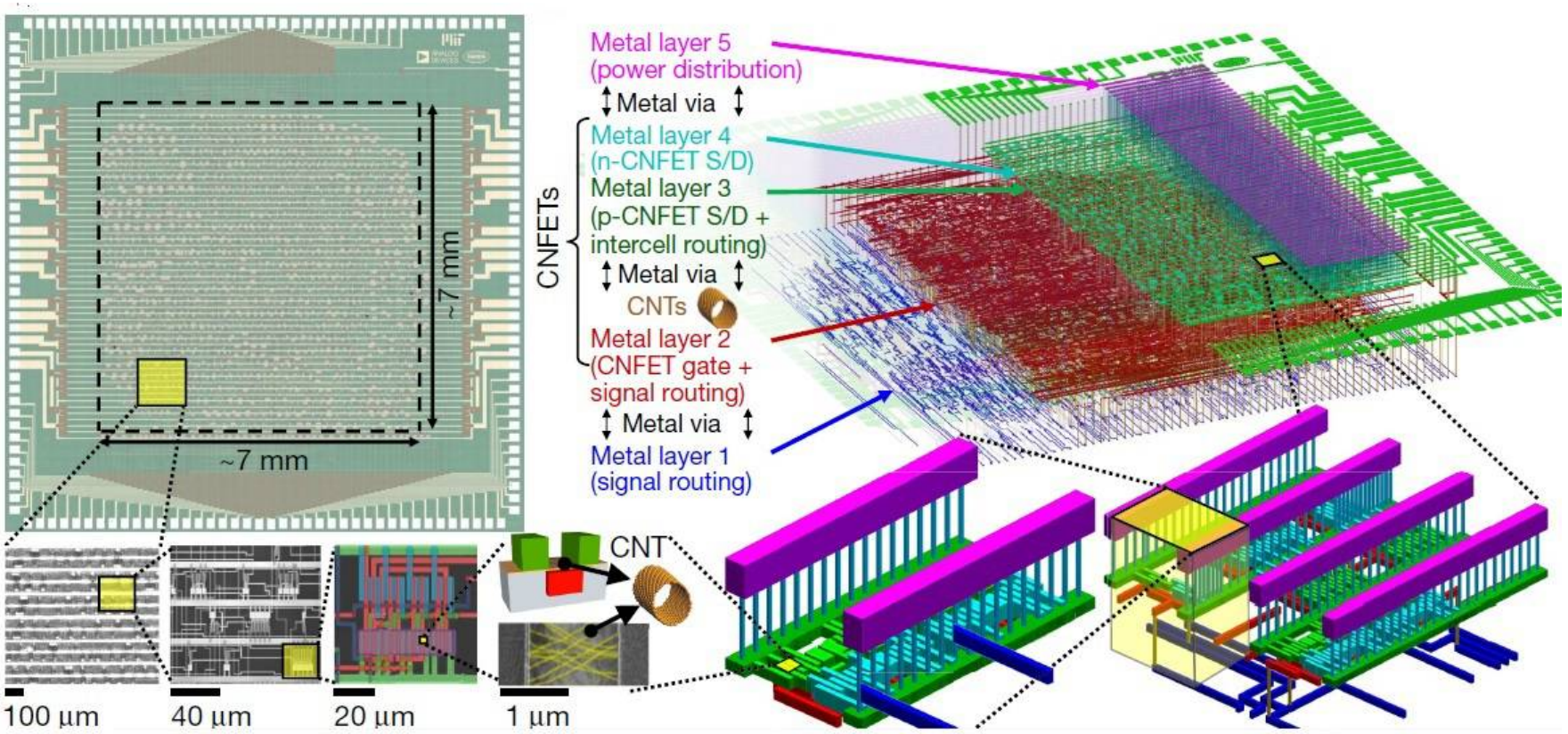






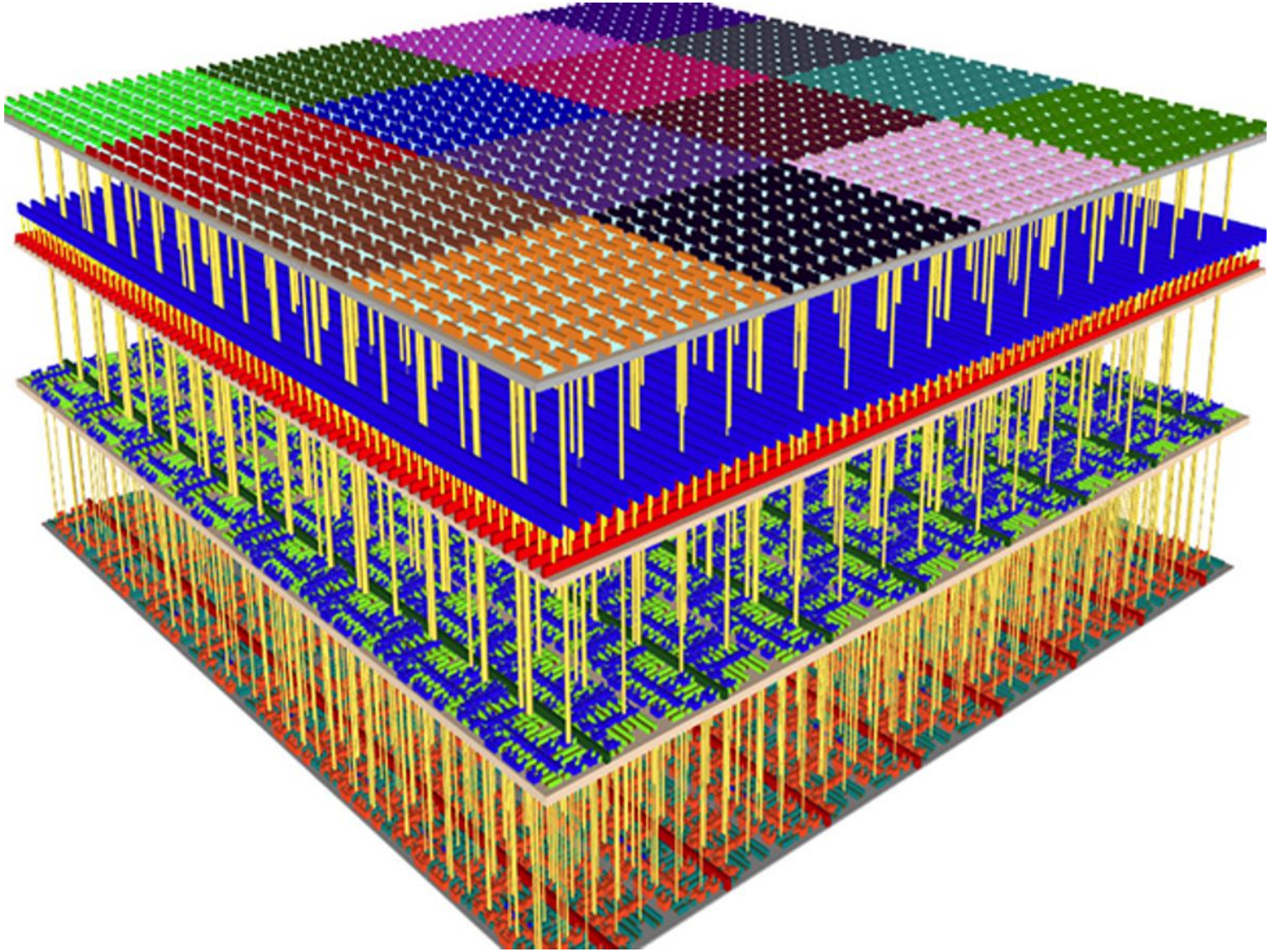
CNT Microprocessor

- CNT are only about 1-2nmin diameter
- Excellent electronic properties



RV16X-NANO. Credit: Nature 572,595–602 (2019)

3D CNT Computer

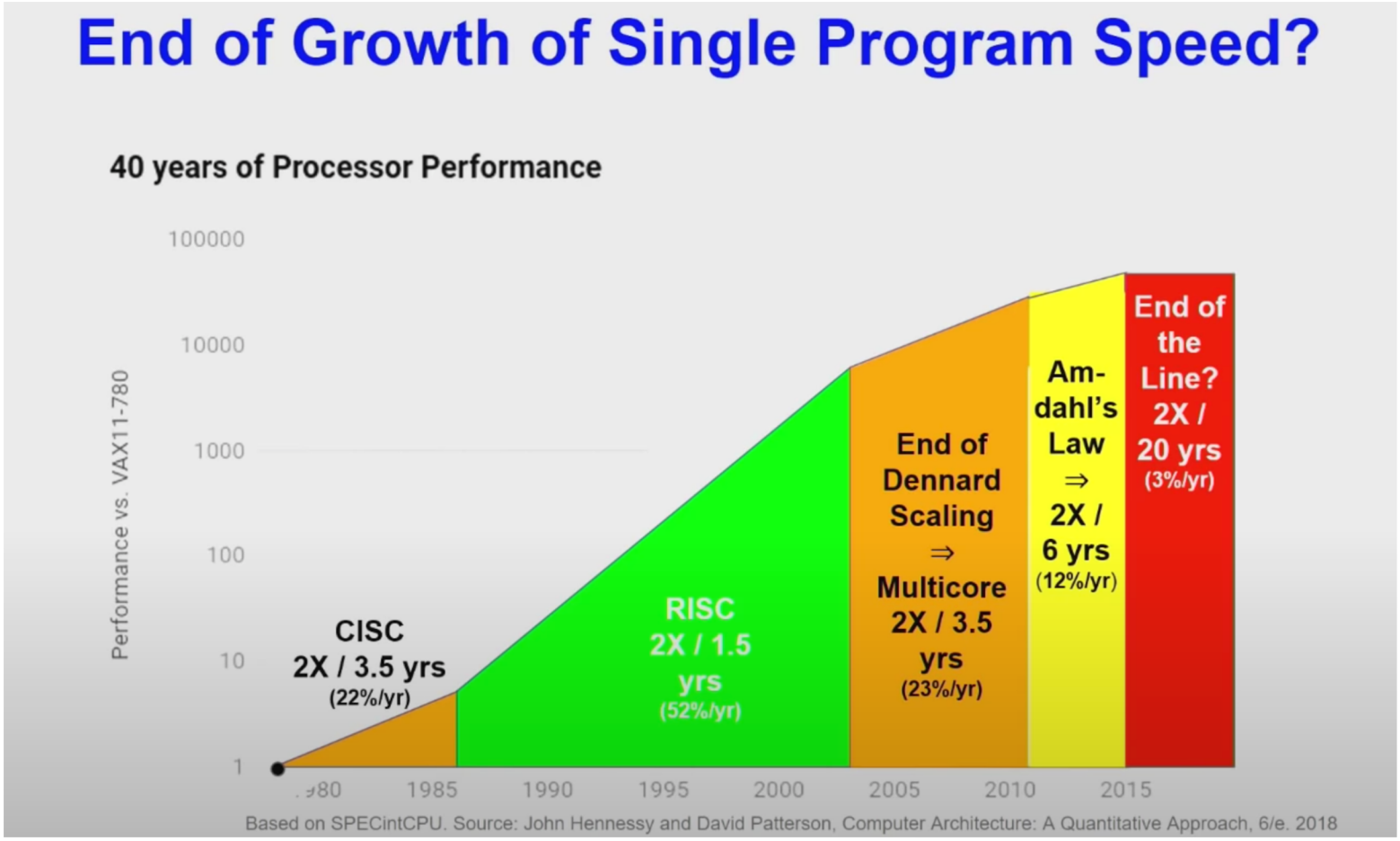


Three-dimensional integration of nanotechnologies for computing and data storage on a single chip

Future of Computing

DOMAIN SPECIFIC ARCHITECTURE

A New Golden Age for Computer Architecture



A New Golden Age for Computer Architecture

What Opportunities Left?

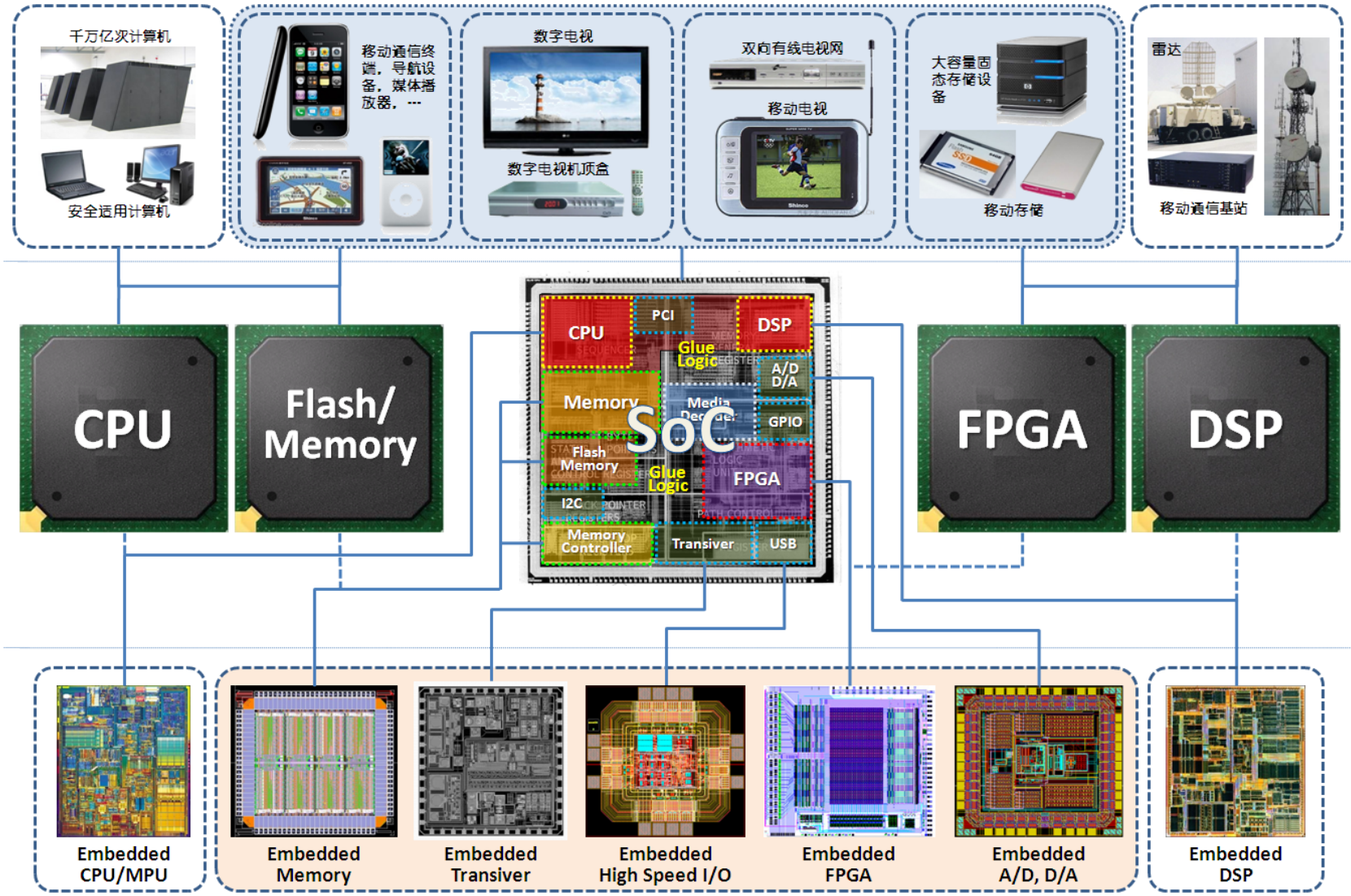
- SW-centric
  - Modern scripting languages are interpreted, dynamically-typed and encourage reuse
  - Efficient for programmers but not for execution
- HW-centric
  - Only path left is *Domain Specific Architectures*
  - Just do a few tasks, but extremely well
- Combination
  - Domain Specific Languages & Architectures

A New Golden Age for Computer Architecture

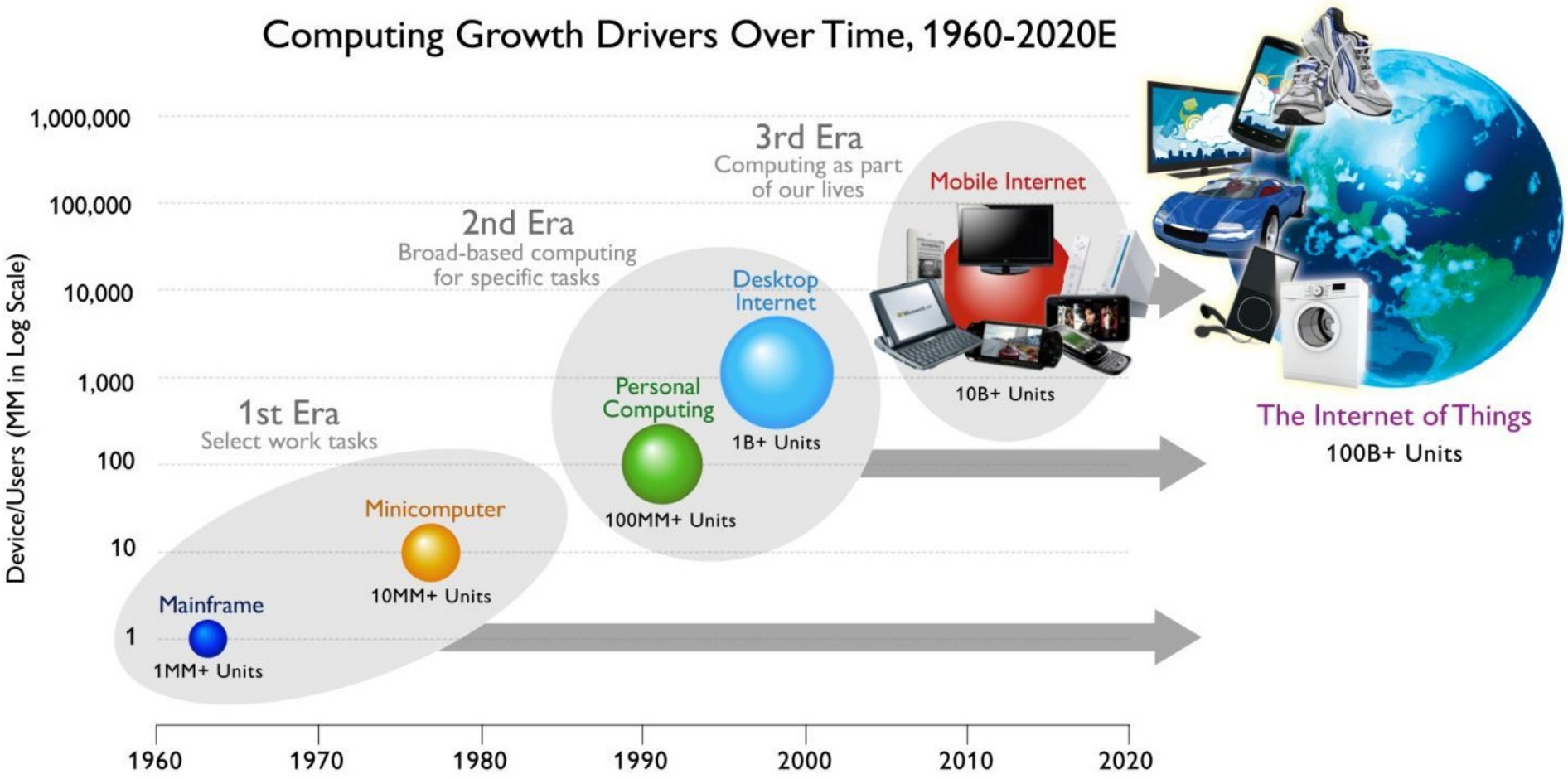
Part III: DSL/DSA Summary

- Lots of opportunities
- But, new approach to computer architecture is needed.
- The Renaissance computer architecture team is vertically integrated. Understands:
  - Applications
  - DSLs and related compiler technology
  - Principles of architecture
  - Implementation technology
- Everything old is new again!

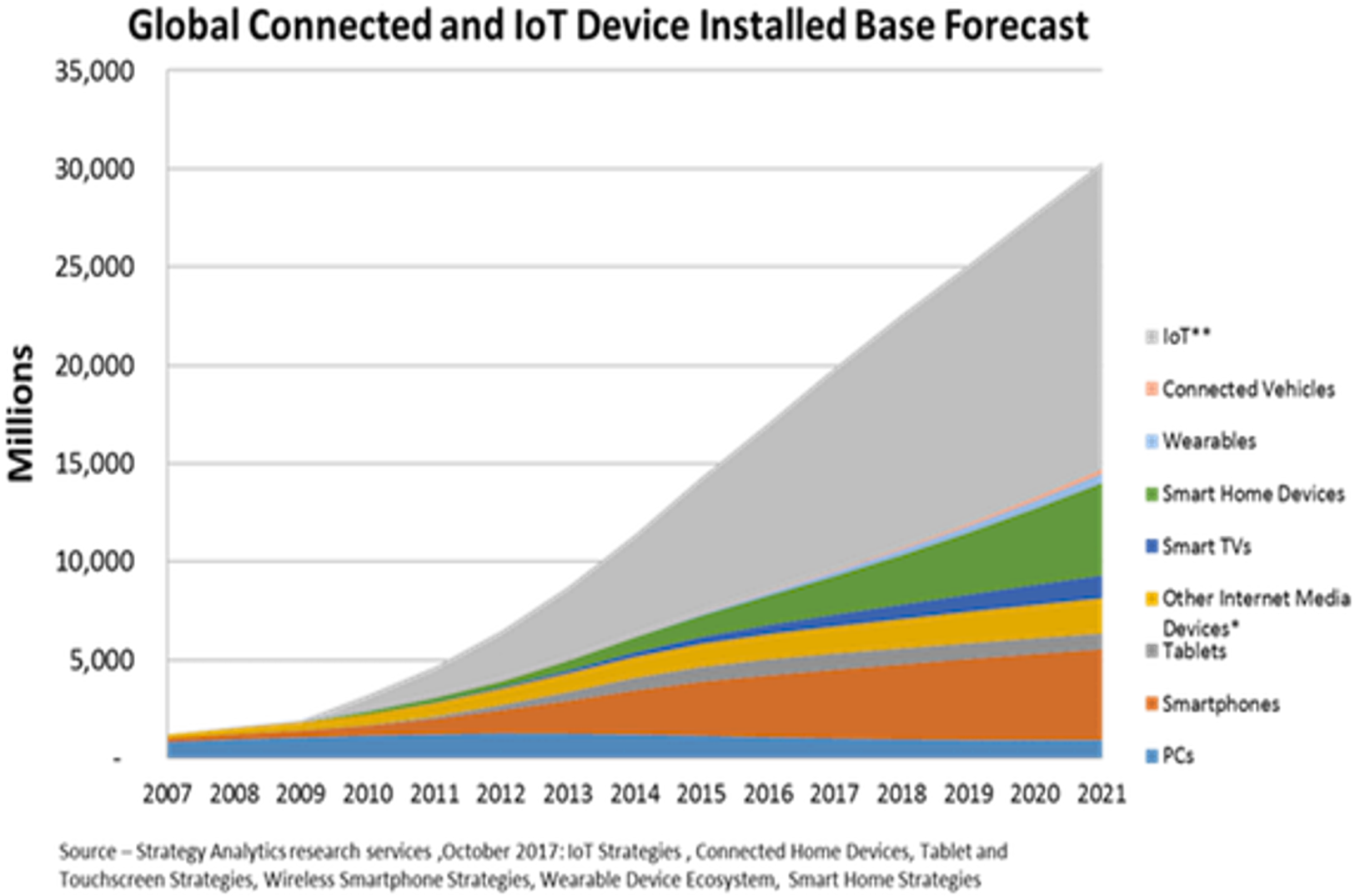
The range of microprocessors is expanding



Post-PC Era


Computing Growth Drivers Over Time, 1960-2020E
Device/Users (MM in Log Scale)
1,000,000
100,000
10,000
1,000
100
10
1
1960
1970
1980
1990
2000
2010
2020
1st Era
Select work tasks
Mainframe
1MM+ Units
Minicomputer
10MM+ Units
2nd Era
Broad-based computing for specific tasks
Personal Computing
100MM+ Units
Desktop Internet
1B+ Units
3rd Era
Computing as part of our lives
Mobile Internet
10B+ Units
The Internet of Things
100B+ Units

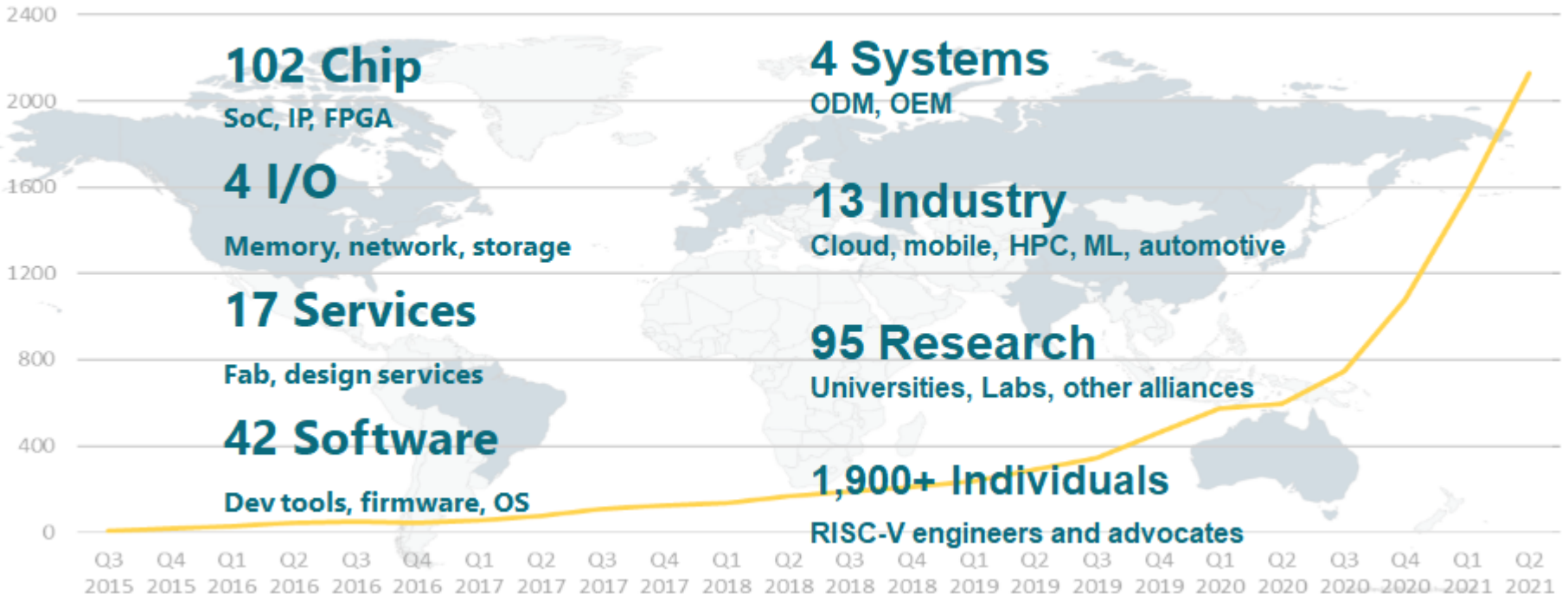
IoT: New growth



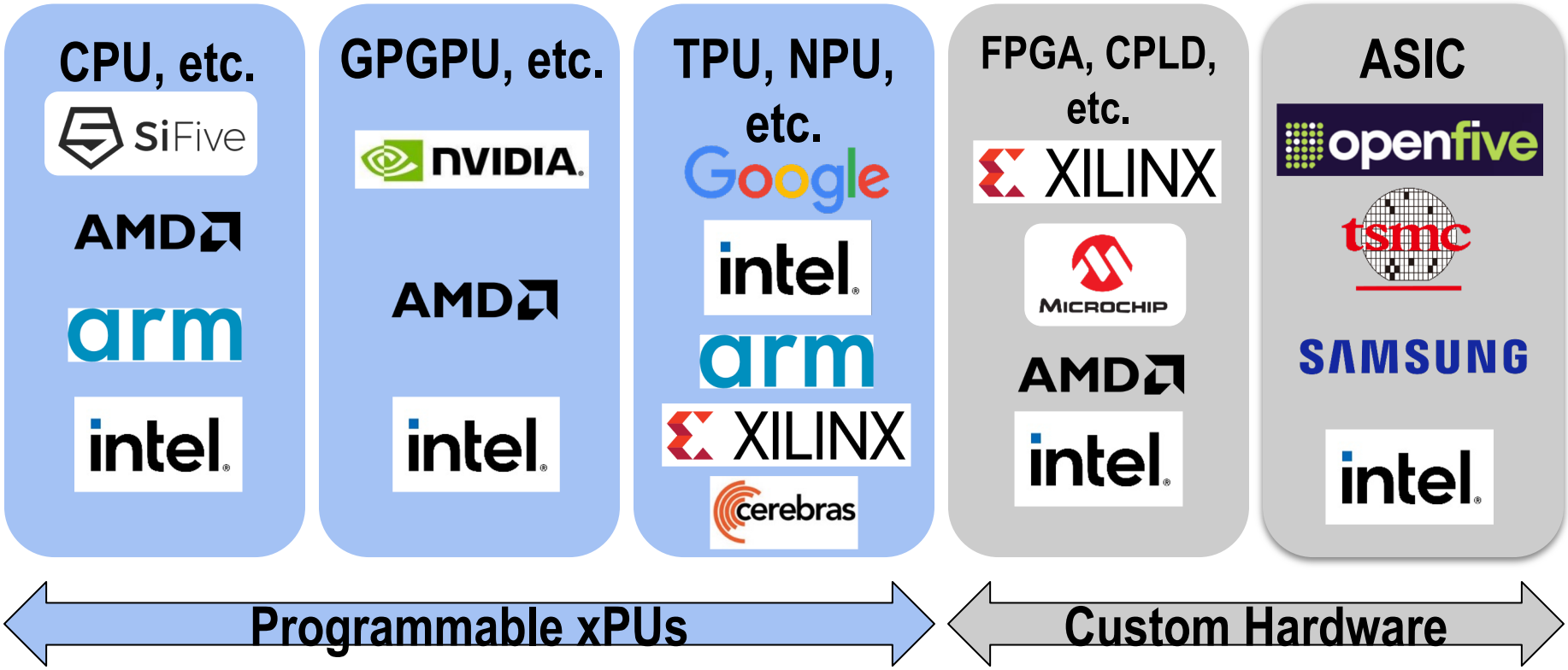
Source – Strategy Analytics research services ,October 2017: IoT Strategies , Connected Home Devices, Tablet and Touchscreen Strategies, Wireless Smartphone Strategies, Wearable Device Ecosystem, Smart Home Strategies

RISC-V??

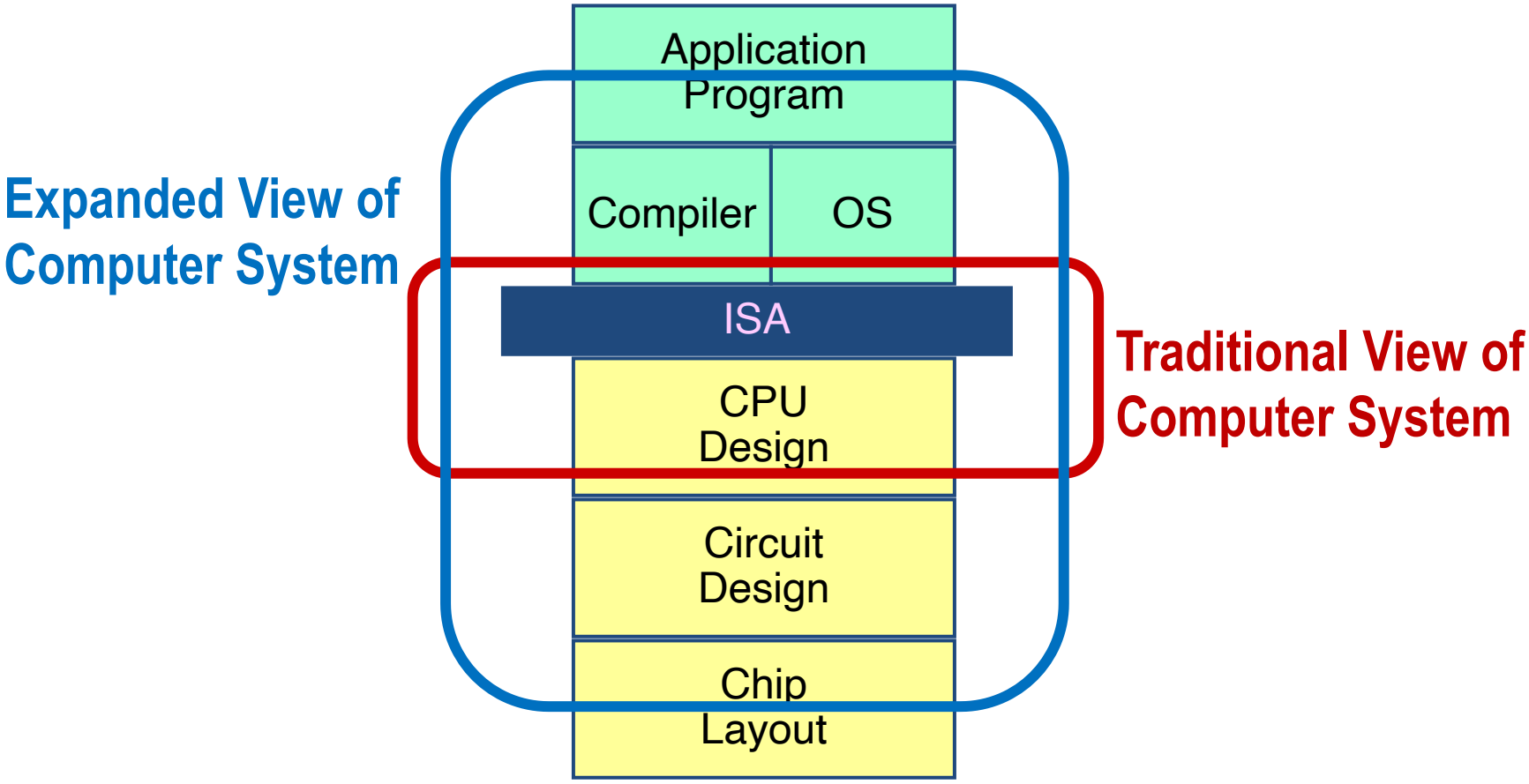
Barriers	Legacy ISA	RISC-V ISA
Complexity	1500+ base instructions Incremental ISA	**47 base instructions** **Modular ISA**
Design freedom	$$$ – Limited	**Free – Unlimited**



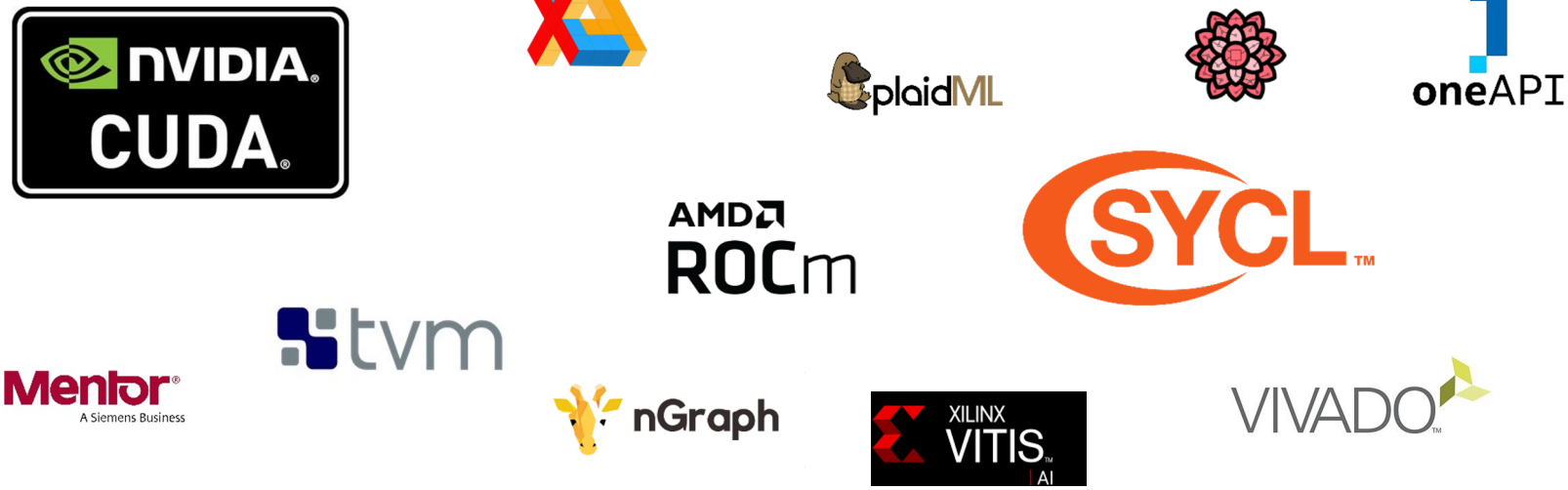
Lots of players! (an incomplete list!)



Expand the View of Computer System



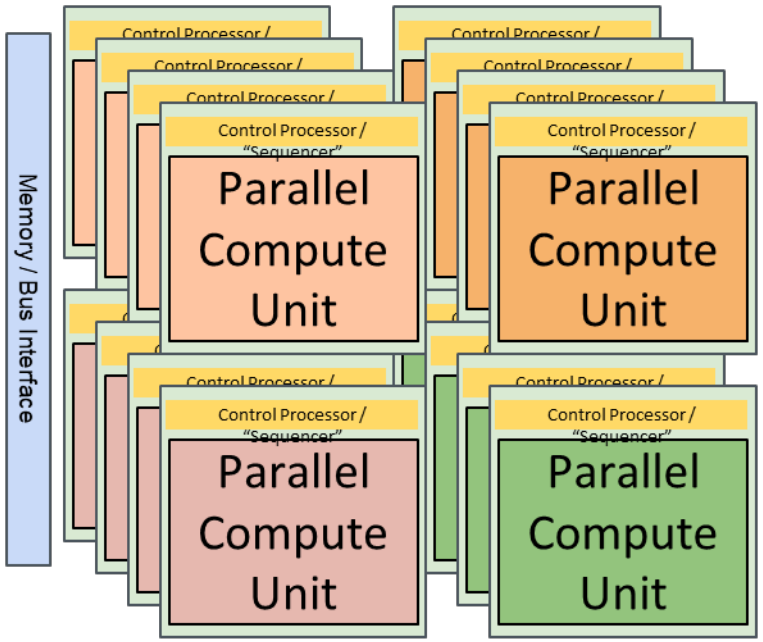
How do we compile/program for this?



⇒ Not very compatible, inconsistent quality and scope
… and don't share much code

"DSA Compilers" to the rescue

Hardware



Software

Programming Model + Userspace API

Accelerator Kernel Compiler

Multistream Mgmt / Interop Parallelism
Memory + Communication Optimization
Heterogenous Device + Host fallback
Kernel Code Generation

Control Proc Assembler + Kernel Driver

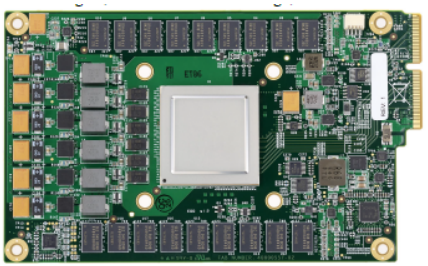
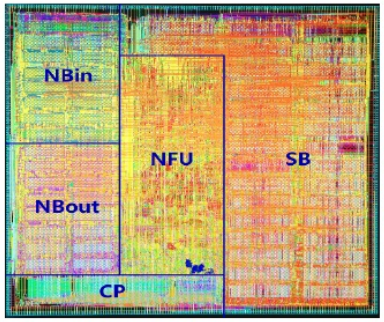
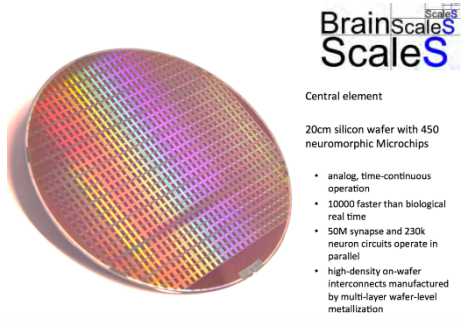


Figure 3. TPU Printed Circuit Board. It can be inserted into the slot for a SATA disk in a server.

Source: Jouppi et al. "In-Datacenter Performance Analysis of a Tensor Processing Unit." (2017)



Source: Tianshi Chen, et. al., "DianNao: A Small-Footprint High-Throughput Accelerator for Ubiquitous Machine-Learning" (2014)



Source: Steve Furber et al., Neuromorphic Computing in the HBP (slides, 2017)

Co-design of HW and SW

- Possible Solution?