Lecture 26:

The Future of High-Performance Computing

Parallel Computer Architecture and Programming

CMU 15-418/15-618, Spring 2017

Comparing Two Large-Scale Systems

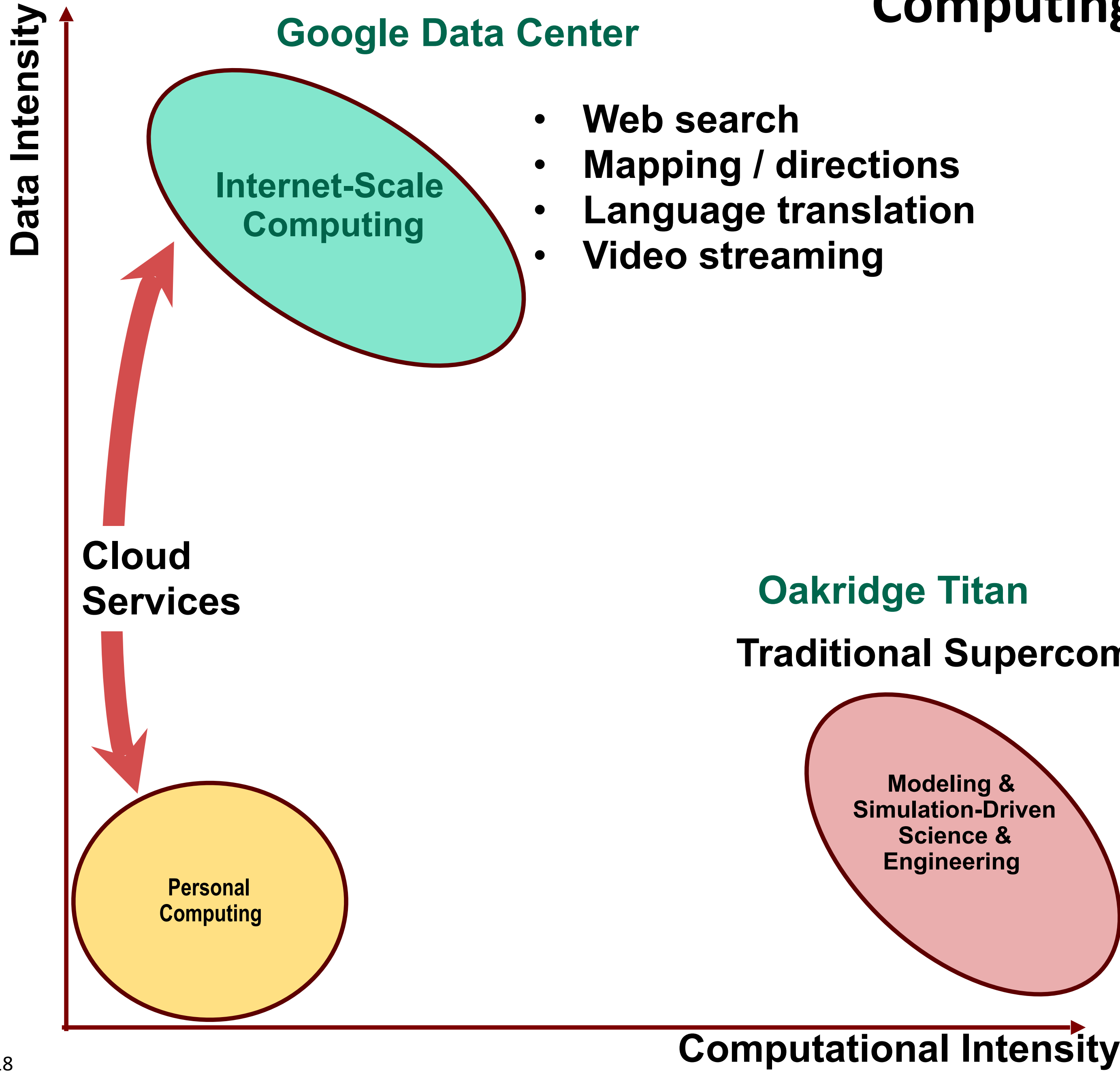
- **Oakridge Titan**

  - Monolithic supercomputer (3rd fastest in world)
  - Designed for compute-intensive applications

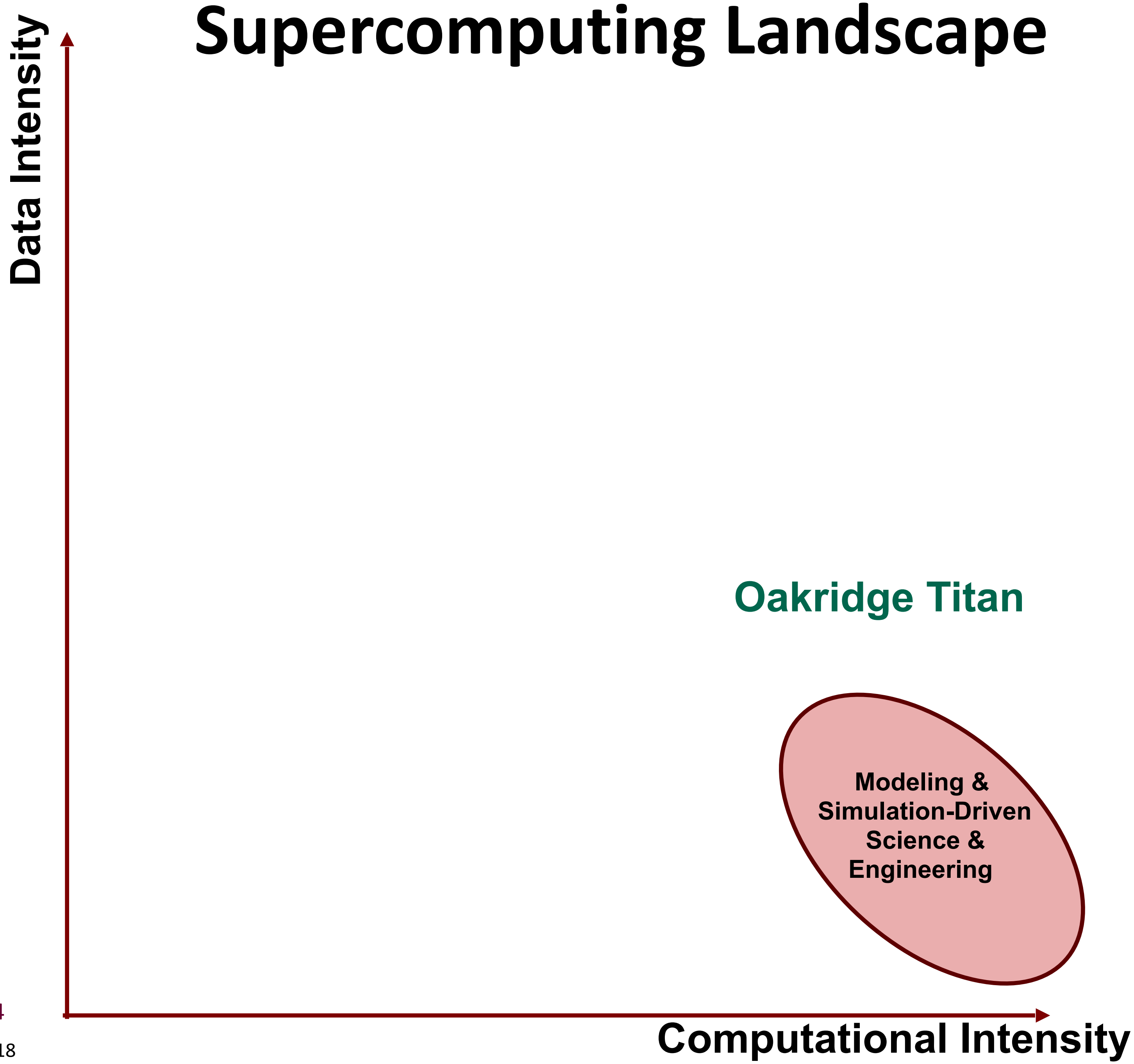
- **Google Data Center**

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

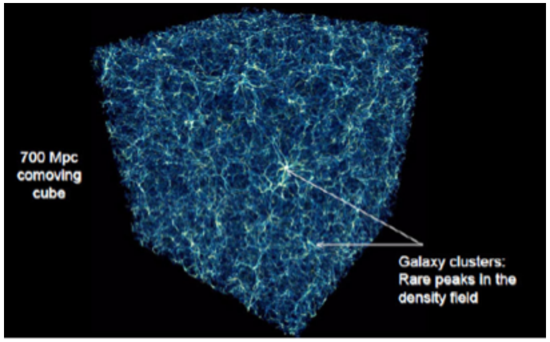
Computing Landscape



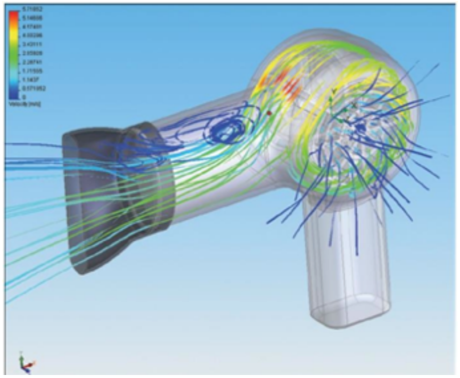
Supercomputing Landscape



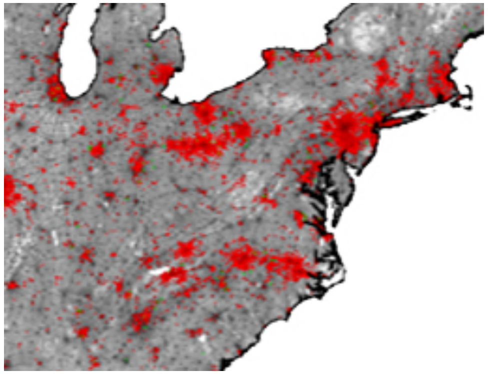
Supercomputer Applications



Science



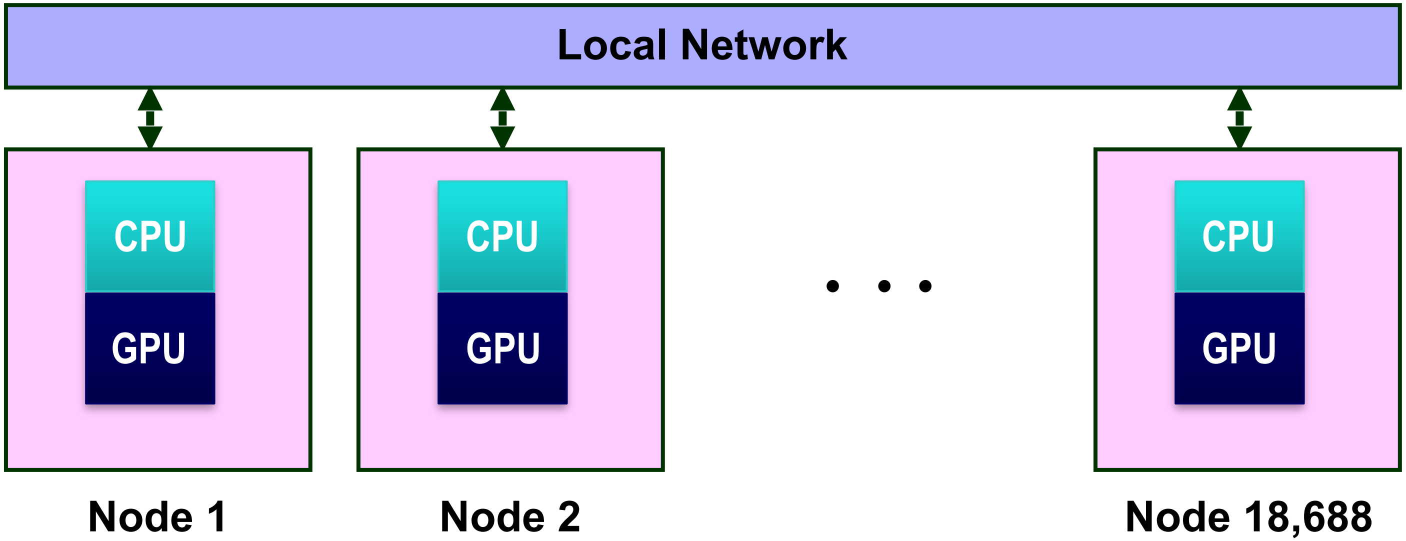
Industrial Products



Public Health

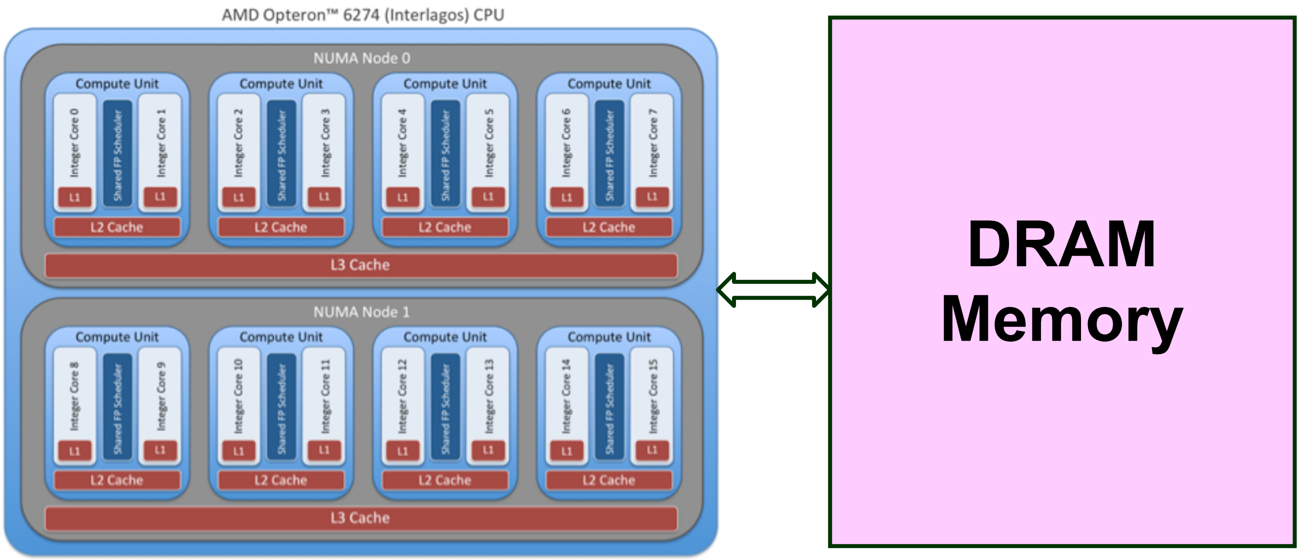
- **Simulation-Based Modeling**
  - System structure + initial conditions + transition behavior
  - Discretize time and space
  - Run simulation to see what happens
- **Requirements**
  - Model accurately reflects actual system
  - Simulation faithfully captures model

Titan Hardware



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

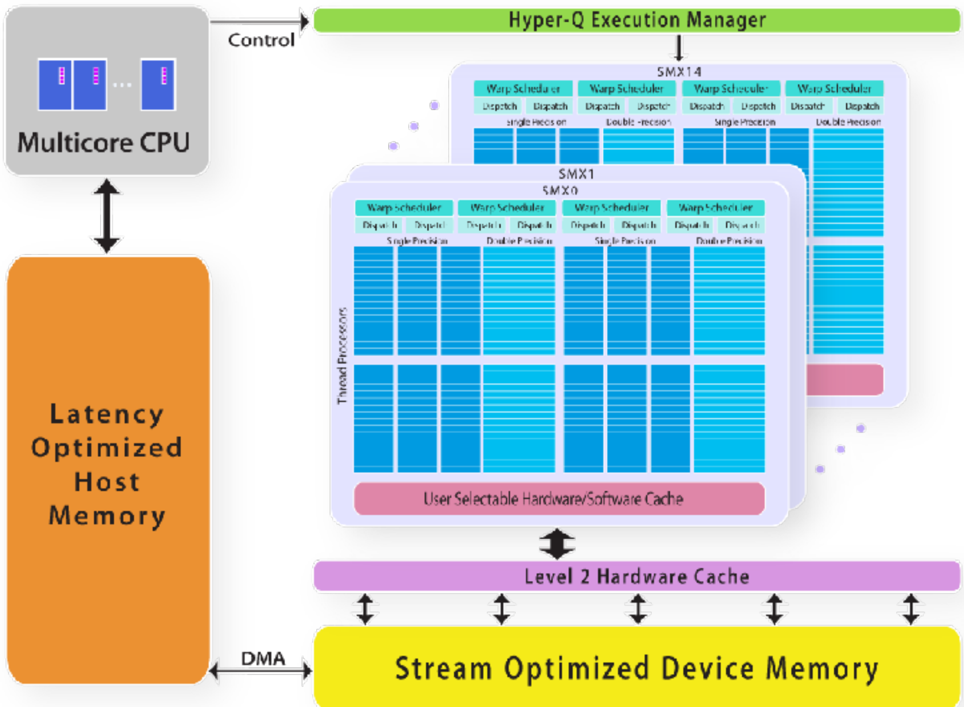
Titan Node Structure: CPU



■ 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

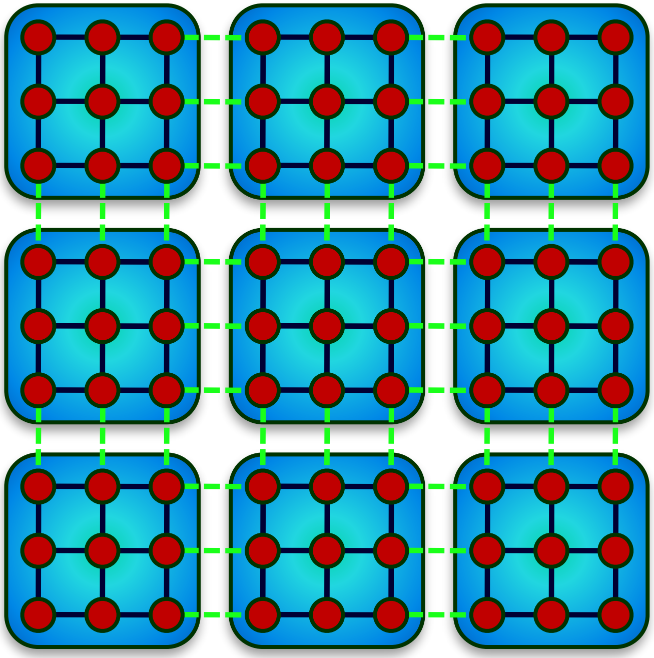


©2013 The Portland Group, Inc.

- **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 10^{12} FLOPS peak performance

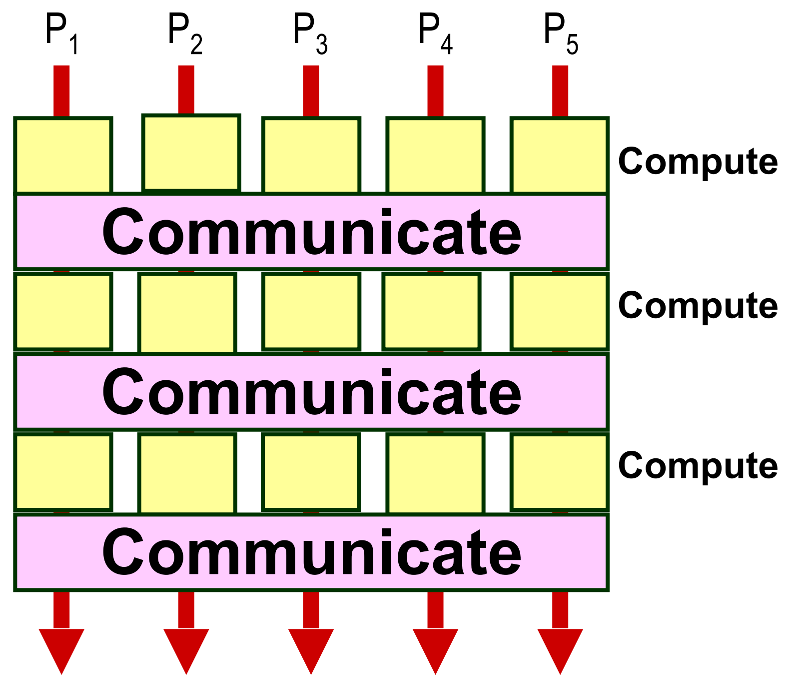
Titan Programming: Principle

- **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

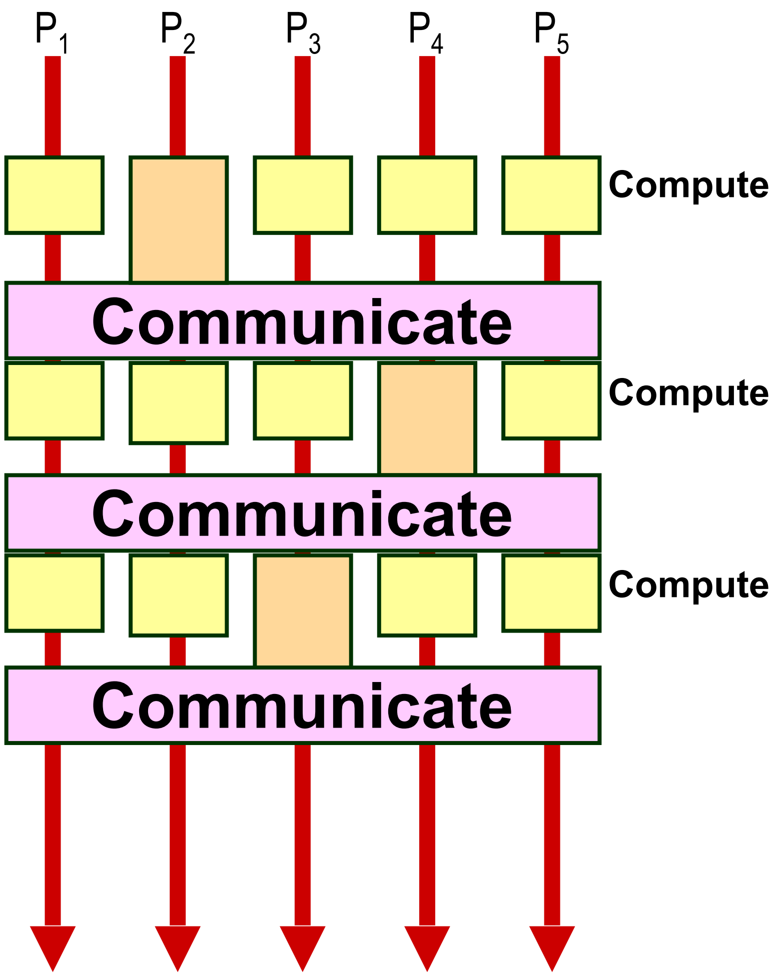


Titan Programming: Principle (cont)

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



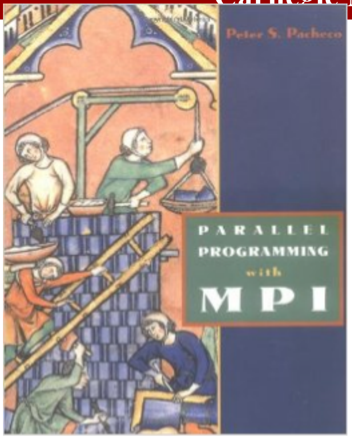
Bulk Synchronous Performance

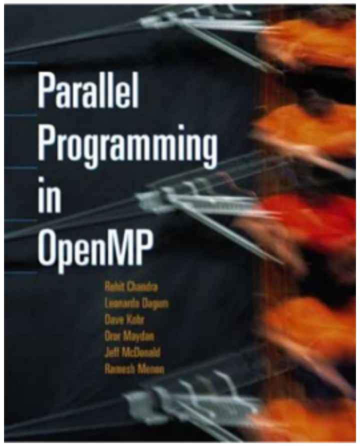


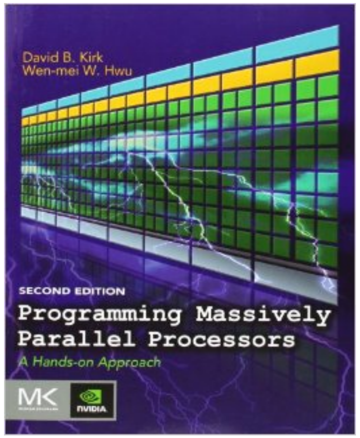
- Limited by performance of slowest processor
- **Strive to keep perfectly balanced**
  - Engineer hardware to be highly reliable
  - Tune software to make as regular as possible
  - Eliminate "noise"
    - Operating system events
    - Extraneous network activity

Titan Programming: Reality

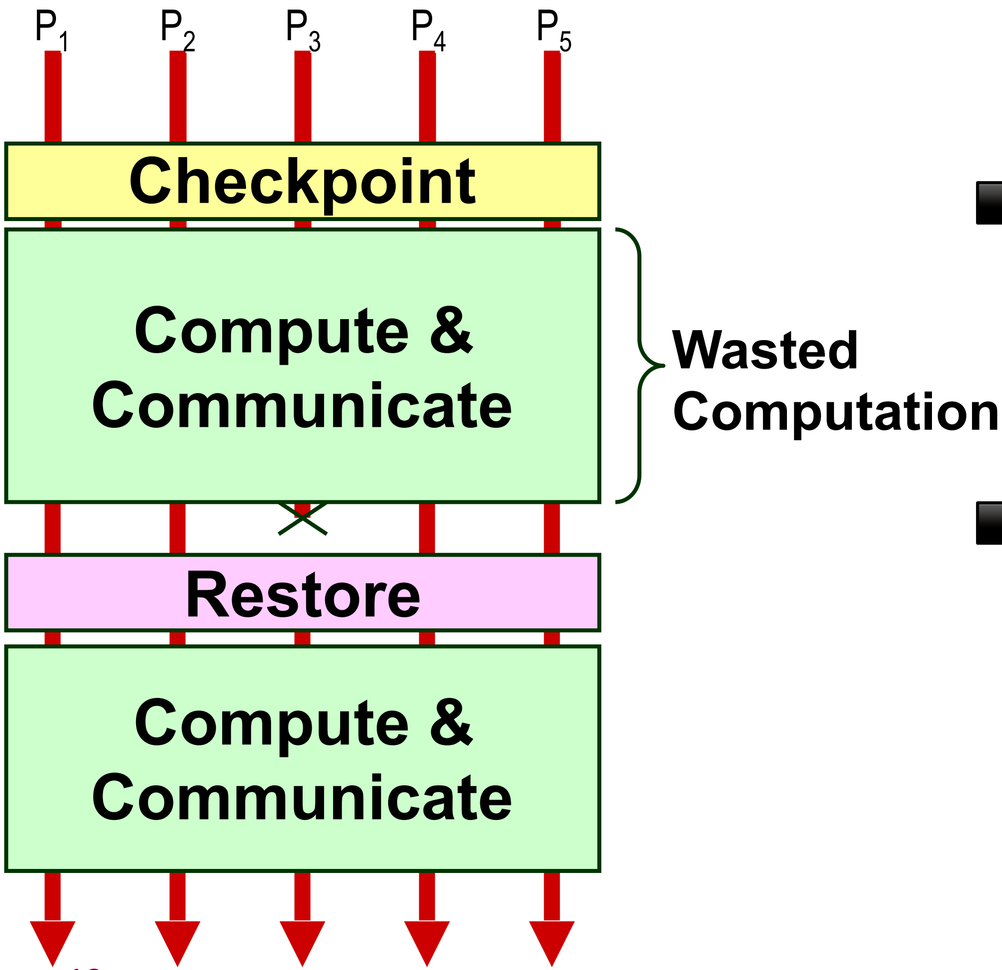
- **System Level**
  - Message-Passing Interface (MPI) supports node computation, synchronization and communication
- **Node Level**
  - OpenMP supports thread-level operation of node CPU
  - CUDA programming environment for GPUs
    - Performance degrades quickly if don't have perfect balance among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





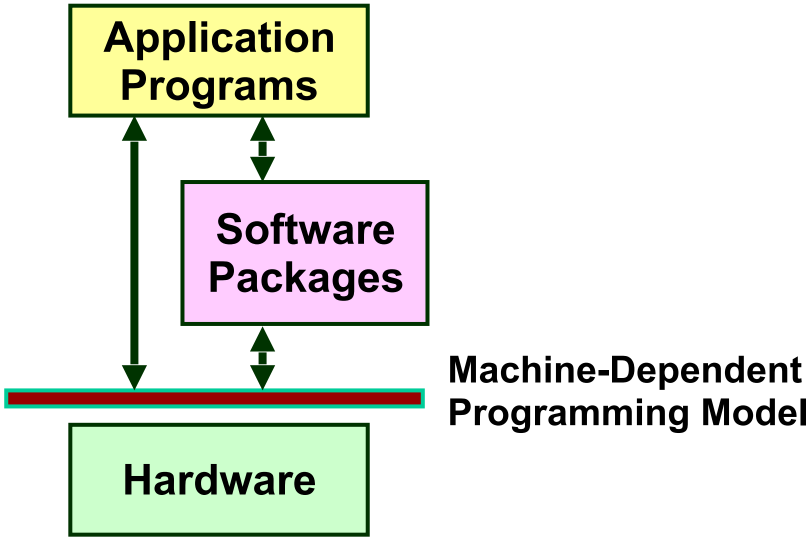


MPI Fault Tolerance



- **Checkpoint**
  - Periodically store state of all processes
  - Significant I/O traffic
- **Restore**
  - When failure occurs
  - Reset state to that of last checkpoint
  - All intervening computation wasted
- **Performance Scaling**
  - Very sensitive to number of failing components

Supercomputer Programming Model



- Program on top of bare hardware

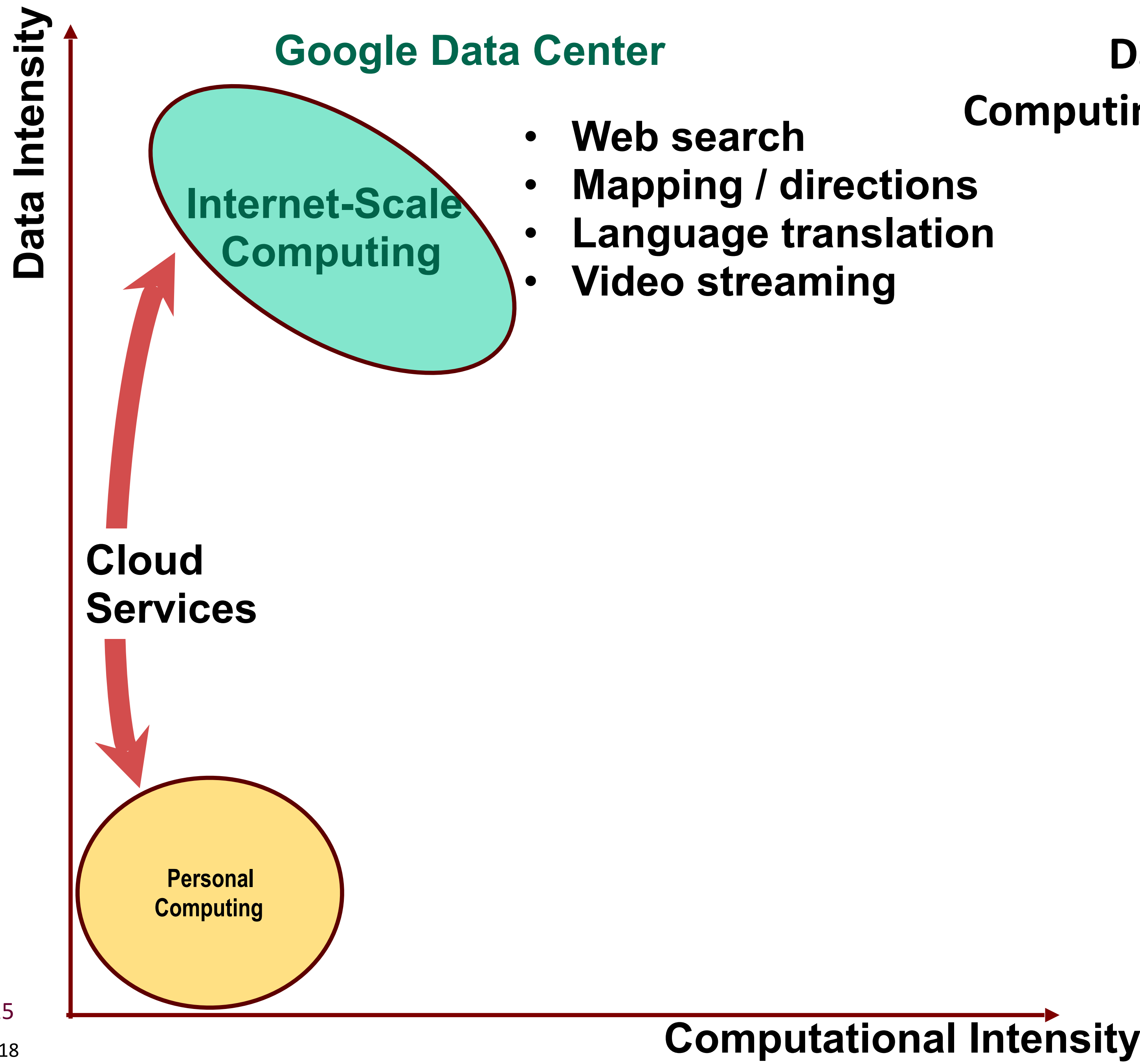
■ Performance

- Low-level programming to maximize node performance
- Keep everything globally synchronized and balanced

■ Reliability

- Single failure causes major delay
- Engineer hardware to minimize failures

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

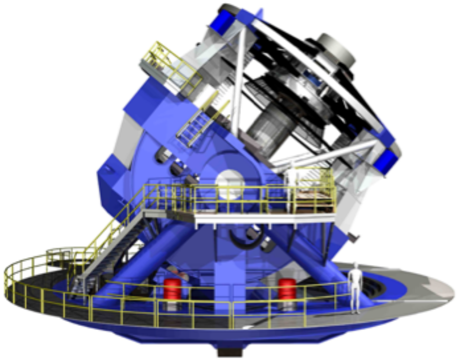
- Must be stored reliably
- Must support real-time updating
- (Relatively) small data volumes

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

Large Synoptic Survey Telescope

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



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

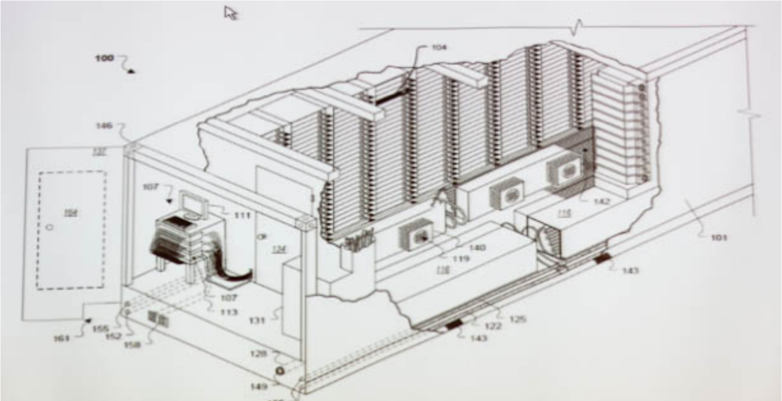


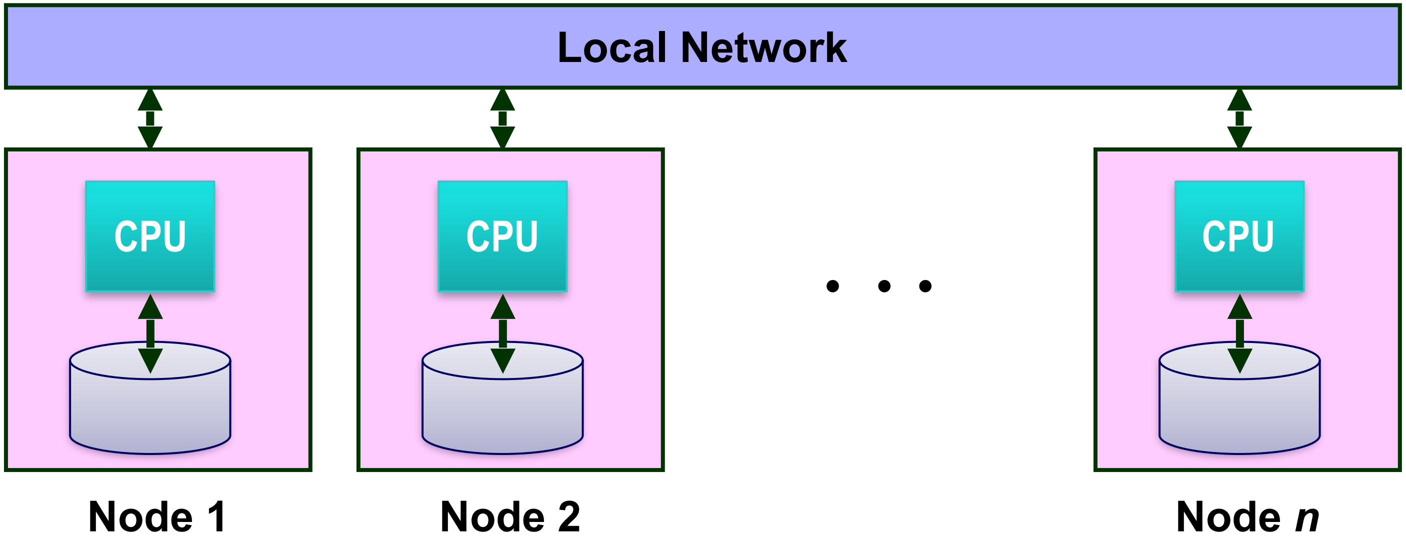
FIG. 1

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

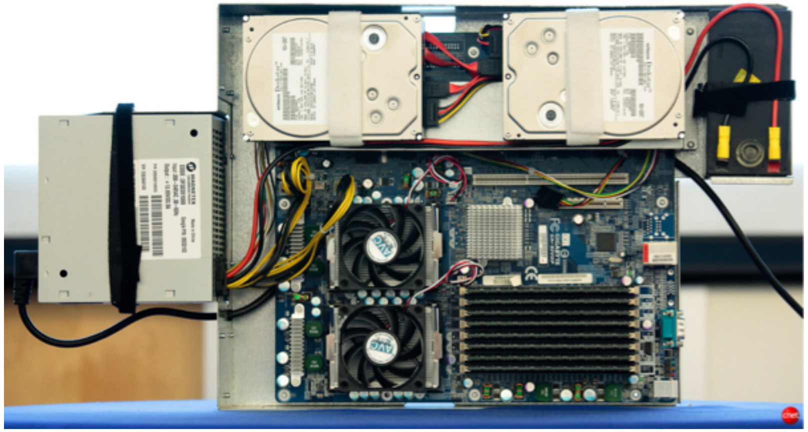
Google Cluster



- Typically 1,000–2,000 nodes

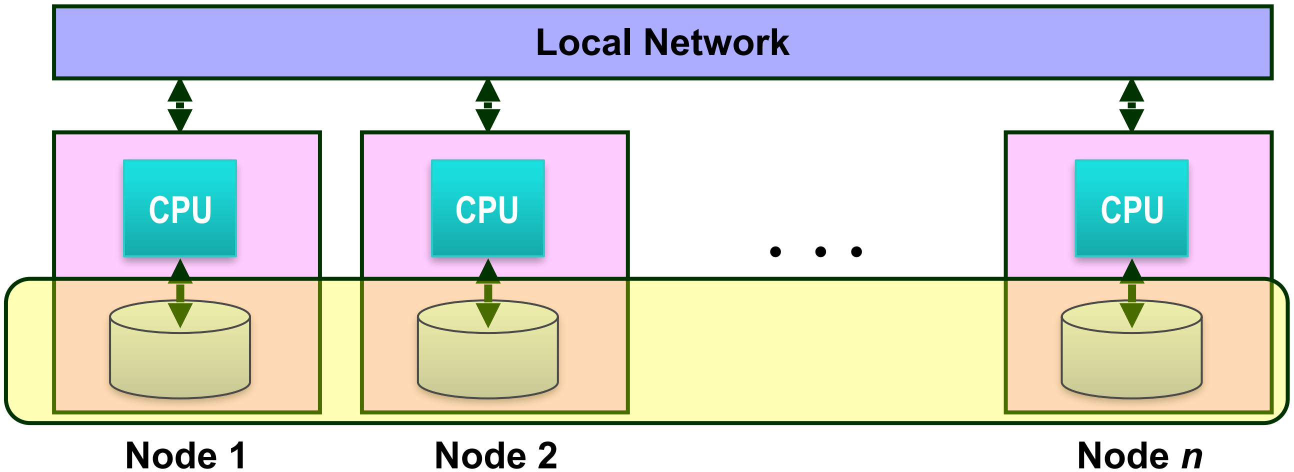
Node Contains

- 2 multicore CPUs
- 2 disk drives
- DRAM



Hadoop Project

- **File system with files distributed across nodes**

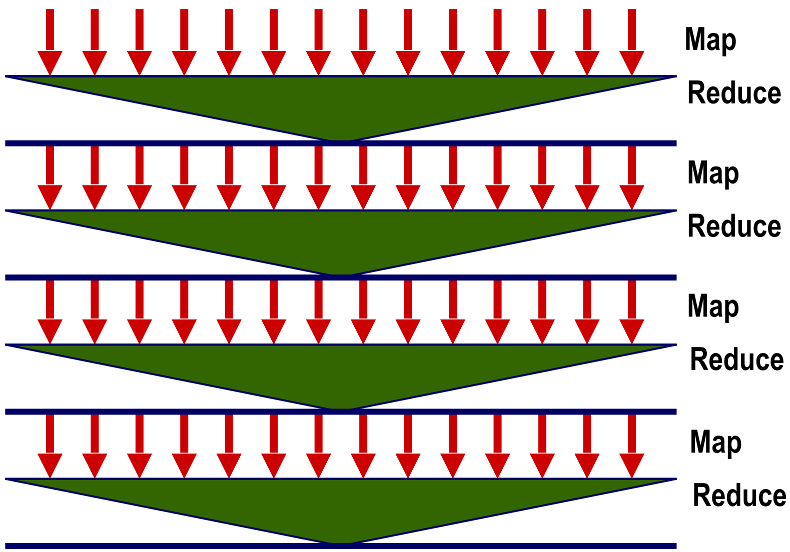


  - Store multiple (typically 3 copies of each file)
    - If one node fails, data still available
  - Logically, any node has access to any file
    - May need to fetch across network

- **Map / Reduce programming environment**
  - Software manages execution of tasks on nodes

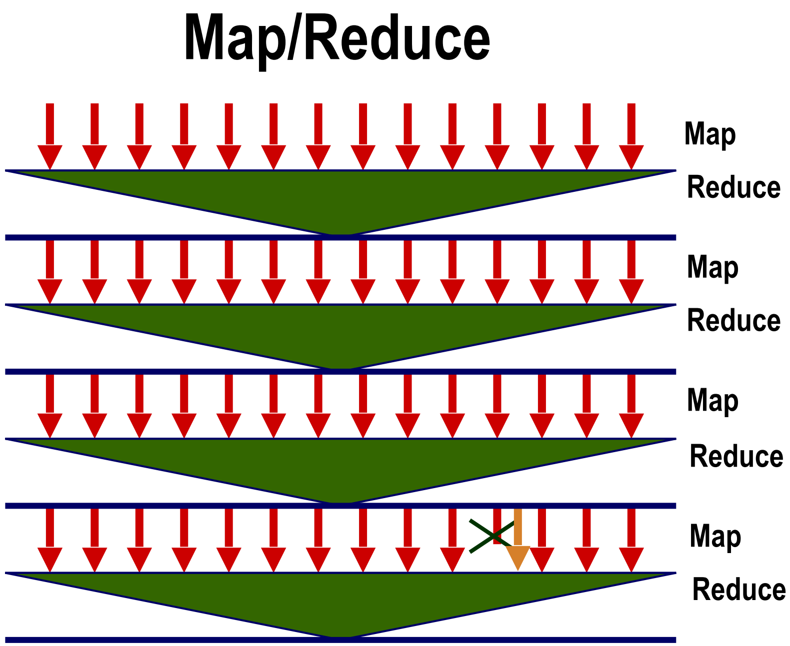
Map/Reduce Operation

Map/Reduce



- **Characteristics**
  - Computation broken into many, short-lived tasks
    - Mapping, reducing
  - Tasks mapped onto processors dynamically
  - Use disk storage to hold intermediate results
- **Strengths**
  - Flexibility in placement, scheduling, and load balancing
  - Can access large data sets
- **Weaknesses**
  - Higher overhead
  - Lower raw performance

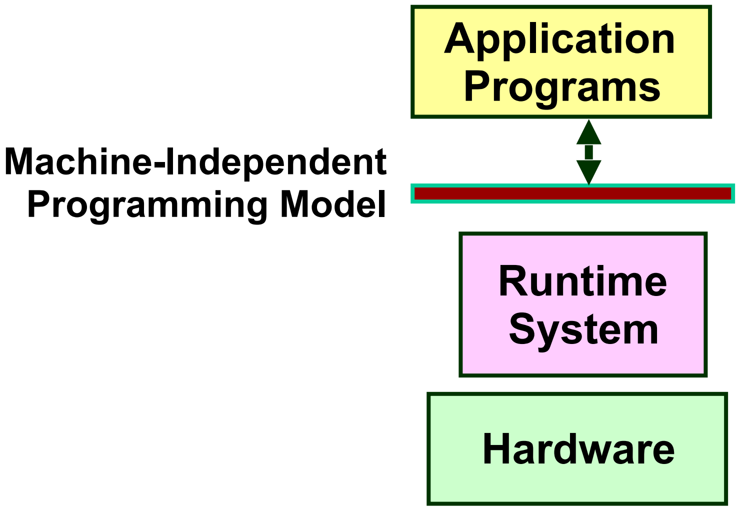
Map/Reduce Fault Tolerance



- **Data Integrity**
  - Store multiple copies of each file
  - Including intermediate results of each Map / Reduce
    - Continuous checkpointing
- **Recovering from Failure**
  - Simply recompute lost result
    - Localized effect
  - Dynamic scheduler keeps all processors busy
- ***Use software to build reliable system on top of unreliable hardware***

Cluster Programming Model

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

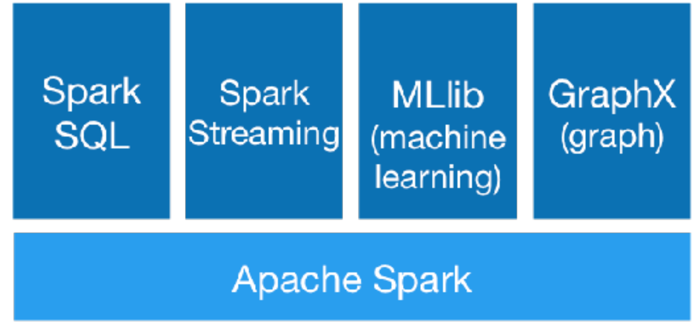


Scaling Challenges

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

Recent Programming Systems

- **Spark Project**



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

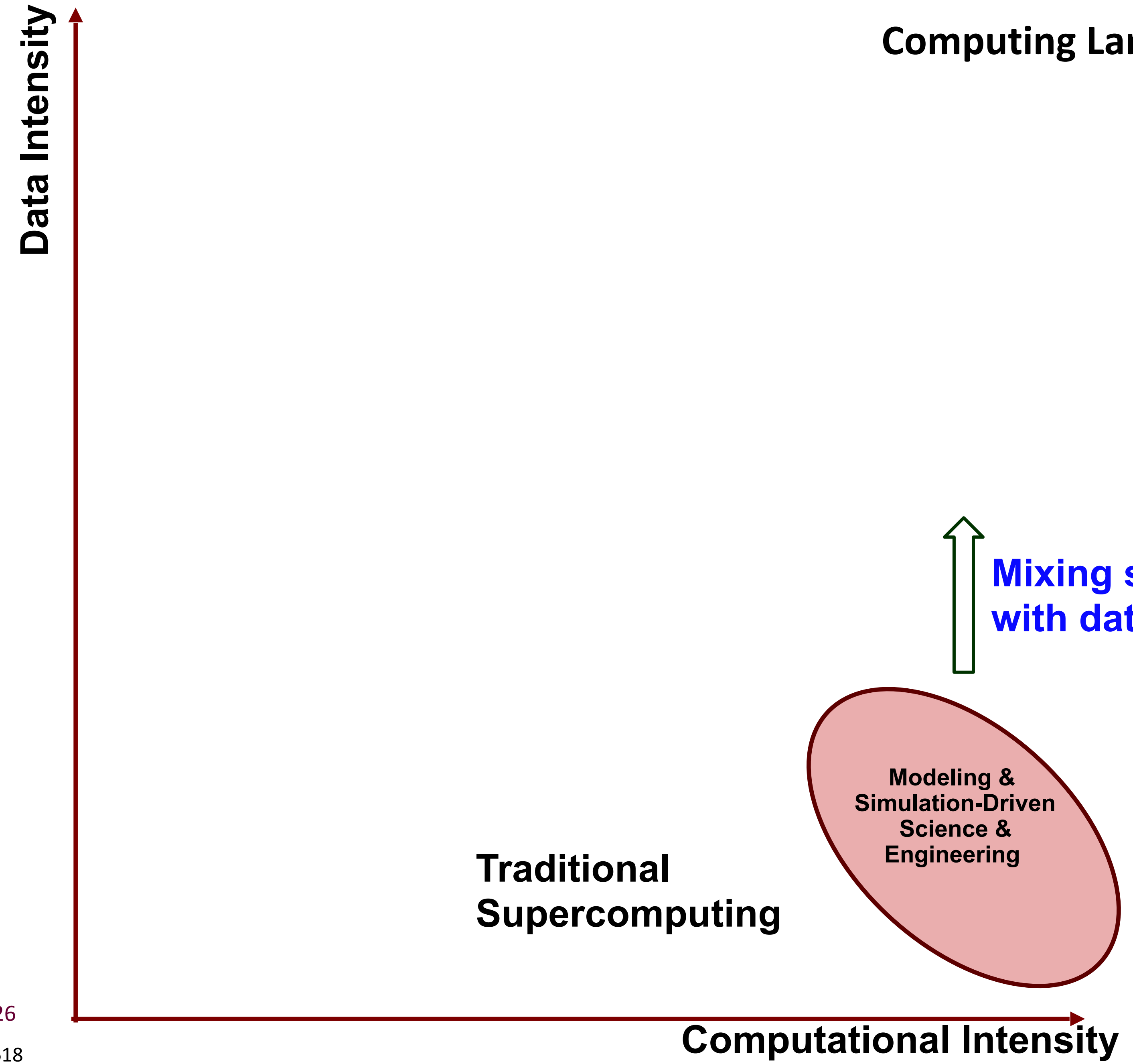
- **GraphLab**


Machine Learning Startup GraphLab Gets A New Name And An $18.5M Check

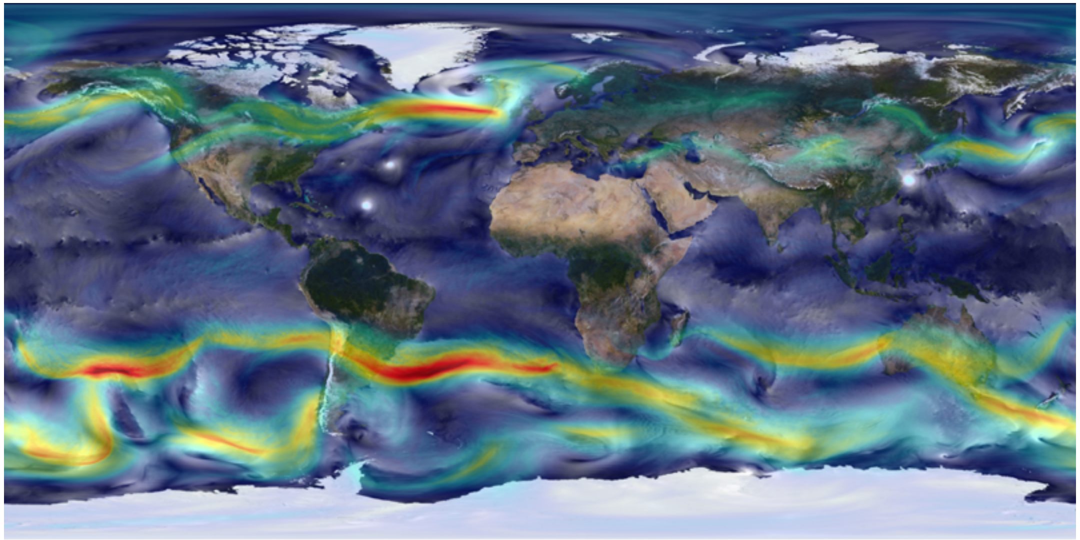
Posted Jan 8, 2015 by Jonathan Shieber (@jshieber)

  - 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



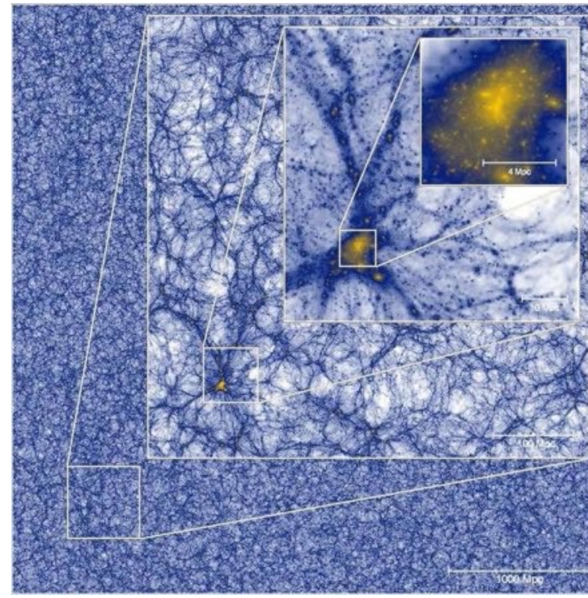
Combining Simulation with Real Data



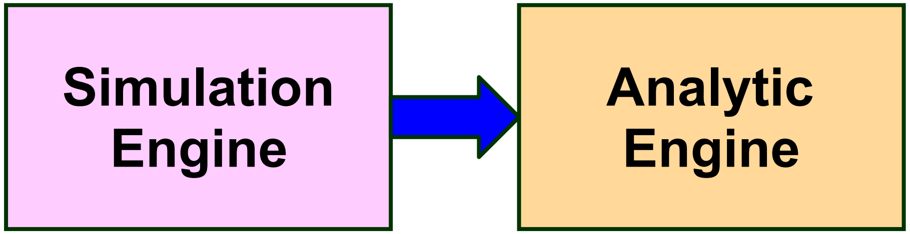
- **Limitations**
  - Simulation alone: Hard to know if model is correct
  - Data alone: Hard to understand causality & "what if"
- **Combination**
  - Check and adjust model during simulation

Real-Time Analytics

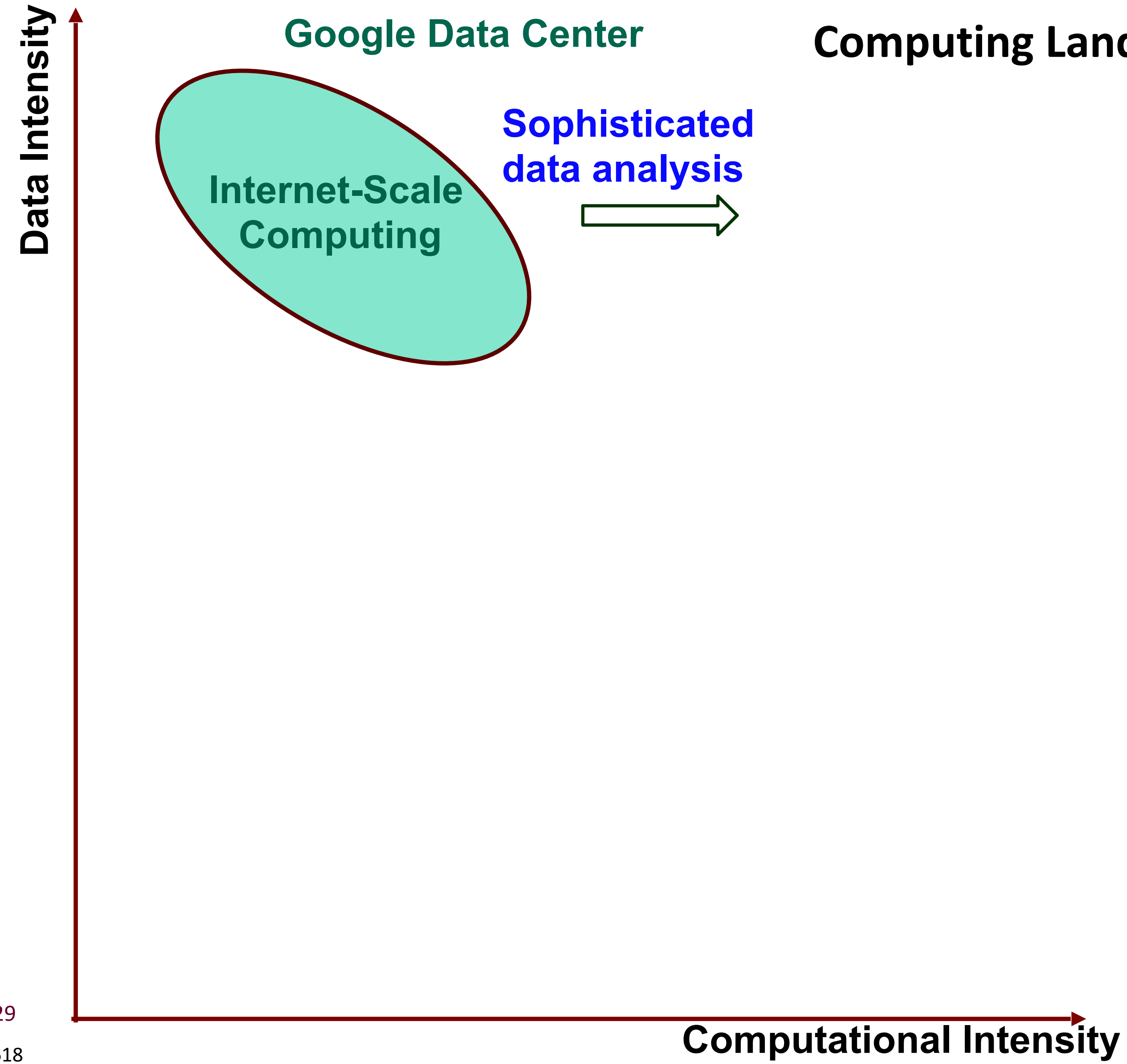
- **Millenium XXL Simulation (2010)**
  - 3 X 10^9 particles
  - Simulation run of 9.3 days on 12,228 cores
  - 700TB total data generated
    - Save at only 4 time points
    - 70 TB
  - Large-scale simulations generate large data sets



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



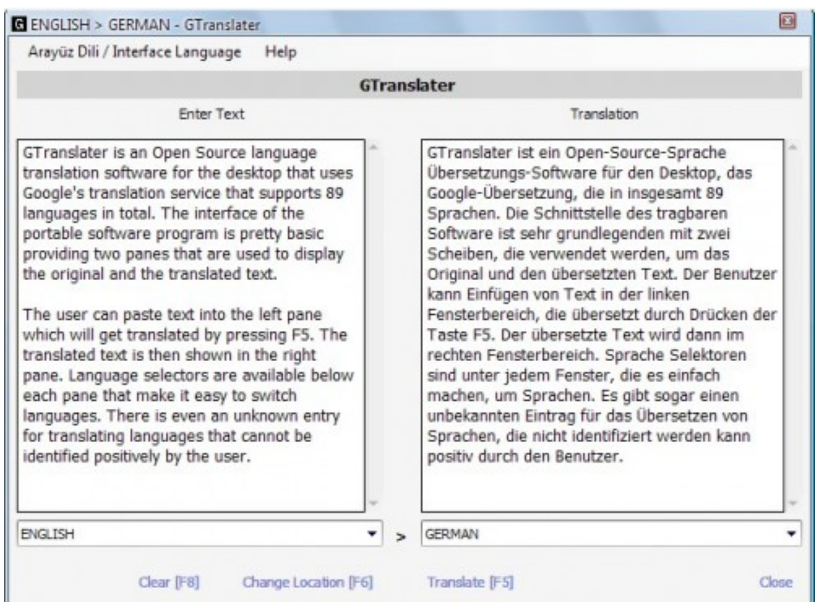
Computing Landscape Trends



Example Analytic Applications

Microsoft Project Adam

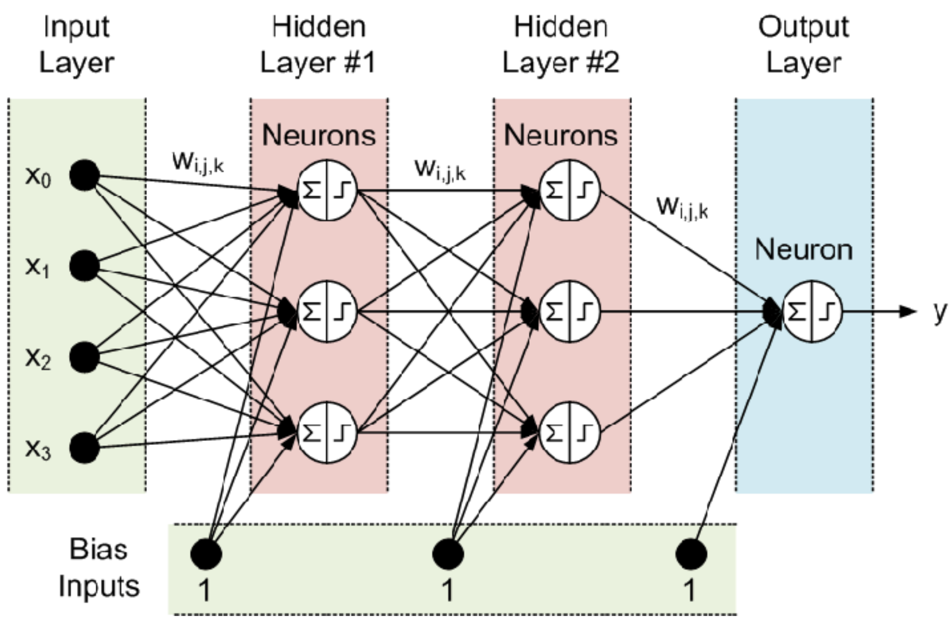
Image → Classifier → Description



English Text → Transducer → German Text

Data Analysis with Deep Neural Networks

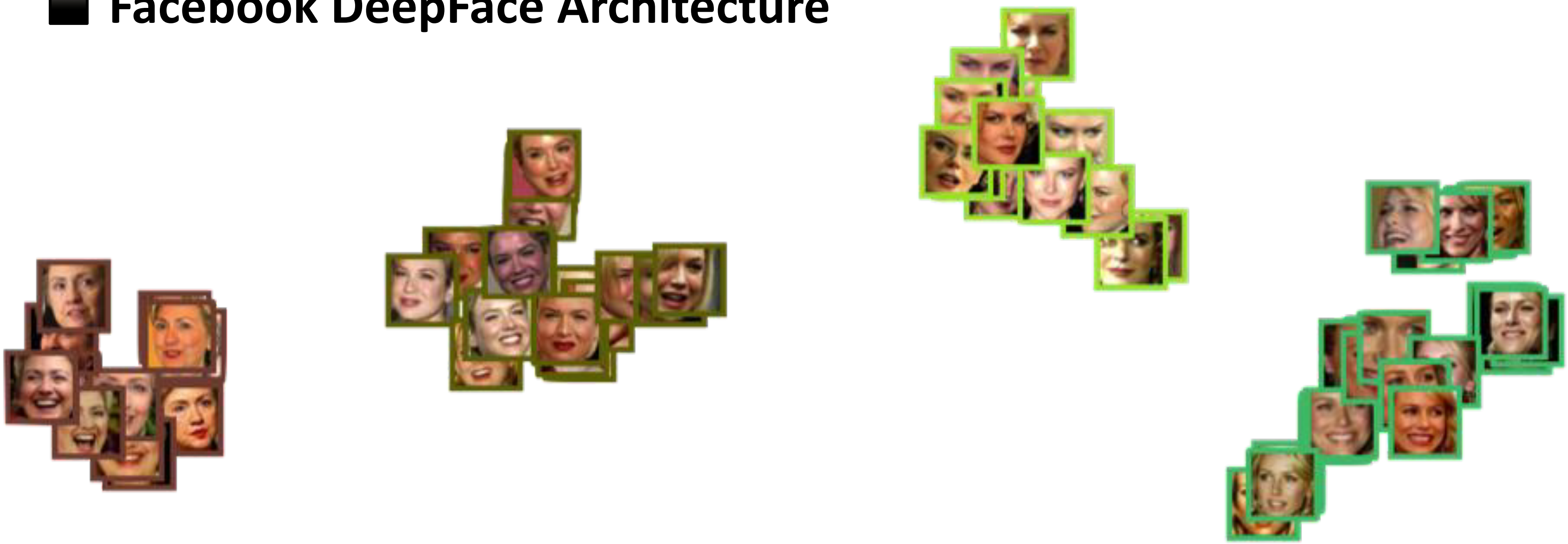
- **Task:**
  - Compute classification of set of input signals

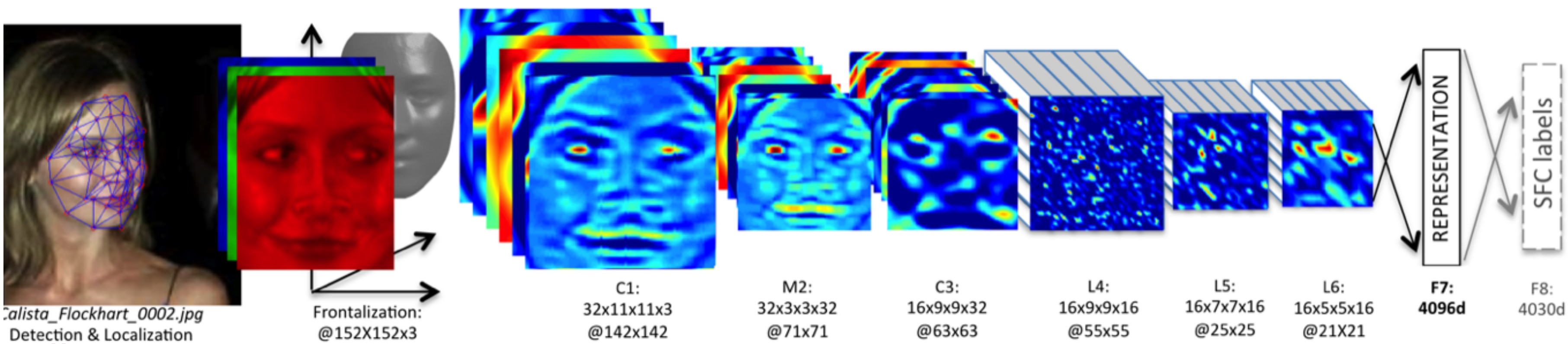


- **Training**
  - **Use many training samples of form input / desired output**
  - **Compute weights that minimize classification error**
- **Operation**
  - **Propagate signals from input to output**

DNN Application Example

- Facebook DeepFace Architecture



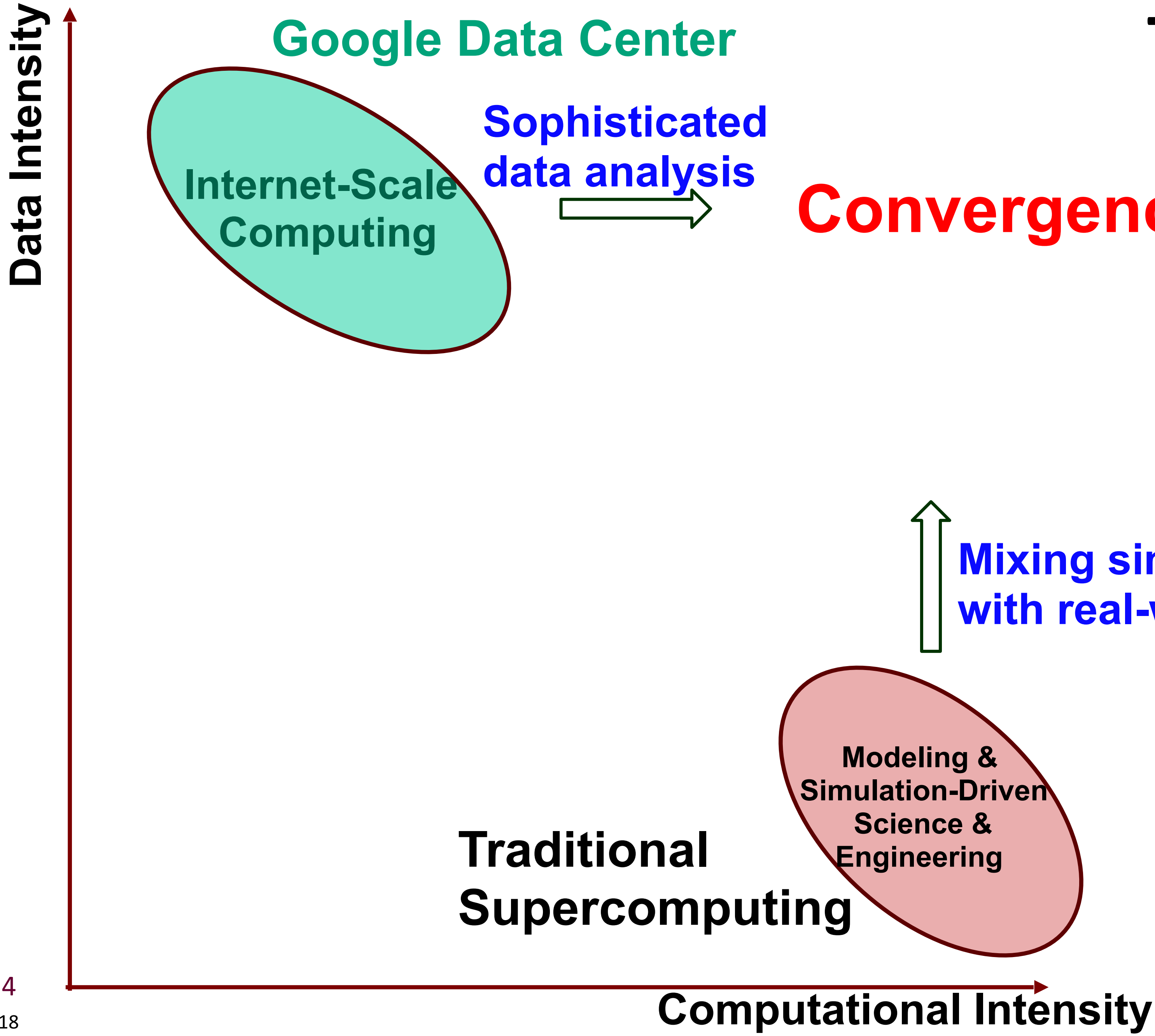


Training DNNs

- **Characteristics**
  - Iterative numerical algorithm
  - Regular data organization

- **Project Adam Training**
  - 2B connections
  - 15M images
  - 62 machines
  - 10 days

Trends


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

Challenges for Convergence

- **Supercomputers**
- **Data Center Clusters**

Hardware

Supercomputers:
- Customized
- Optimized for reliability

Data Center Clusters:
- Consumer grade
- Optimized for low cost

Run-Time System

Supercomputers:
- Source of "noise"
- Static scheduling

Data Center Clusters:
- Provides reliability
- Dynamic allocation

Application Programming

Supercomputers:
- Low-level, processor-centric model

Data Center Clusters:
- High level, data-centric model

Summary: 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 data-centric 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

GETTING TO EXASCALE

World's Fastest Machines

- **Top500 Ranking: High-performance LINPACK**
  - Benchmark: Solve N x N linear system
  - Some variant of Gaussian elimination
    - $2/3\ N^3 + O(N^2)$ operations
  - Vendor can choose N to give best performance (in FLOPS)
- **Alternative: High-performance conjugate gradient**
  - Solve sparse linear system (≤ 27 nonzeros / row)
  - Iterative method
  - Higher communication / compute ratio

Sunway TaihuLight

- **Wuxi China**
  - Operational 2016
- **Machine**
  - Total machine has 40,960 processor chips
  - Processor chip contains 256 compute cores + 4 management cores
  - Each has 4-wide SIMD vector unit
  - 8 FLOPS / clock cycle
- **Performance**
  - HPL: 93.0 PF (World's top)
  - HPCG: 0.37 PF
  - 15.4 MW
  - 1.31 PB DRAM
- **Ratios (Big is Better)**
  - GigaFLOPS/Watt: 6.0
  - Bytes/FLOP: 0.014

Tianhhe-2

- **Guangzhou China**
  - Operational 2013
- **Machine**
  - Total machine has 16,000 nodes
  - Each with 2 Intel Xeons + 3 Intel Xeon Phi's
- **Performance**
  - HPL: 33.9 PF
  - HPCG: 0.58 PF (world's best)
  - 17.8 MW
  - 1.02 PB DRAM
- **Ratios (Big is Better)**
  - GigaFLOPS/Watt: 1.9
  - Bytes/FLOP: 0.030

Titan

- **Oak Ridge, TN**
  - Operational 2012
- **Machine**
  - Total machine has 18,688 nodes
  - Each with 16-core Opteron + Tesla K20X GPU
- **Performance**
  - HPL: 17.6 PF
  - HPCG: 0.32 PF
  - 8.2 MW
  - 0.71 PB DRAM
- **Ratios (Big is Better)**
  - GigaFLOPS/Watt: 2.2
  - Bytes/FLOP: 0.040

How Powerful is a Titan Node?

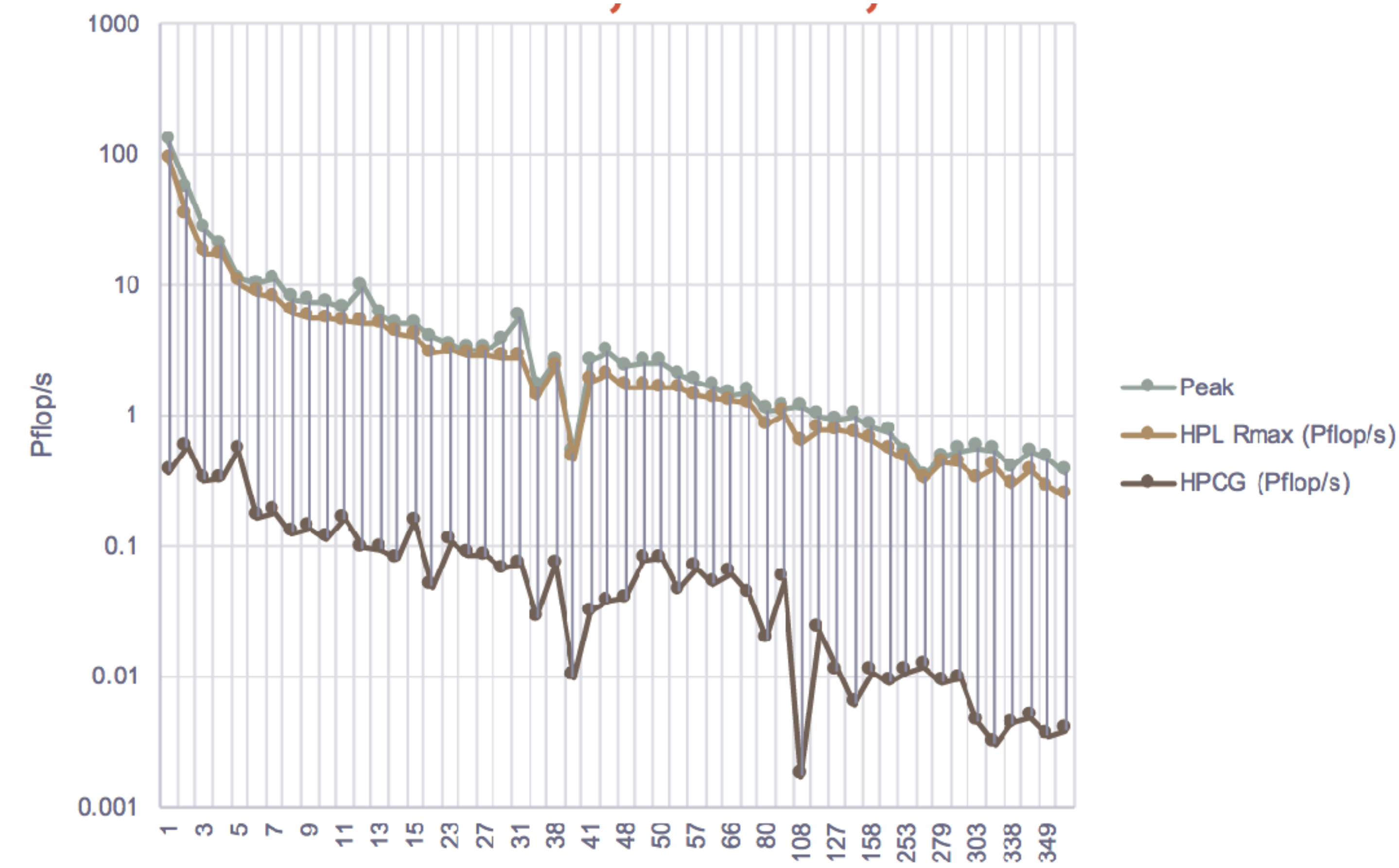
Titan

- **CPU**
  - Opteron 6274
  - Nov., 2011. 32nm technology
  - 2.2 GHz
  - 16 cores (no hyperthreading)
  - 16 MB L3 cache
  - 32 GB DRAM
- **GPU**
  - Kepler K20X
  - Feb., 2013. 28nm
  - Cuda capability 3.5
  - 3.9 TF Peak (SP)

GHC Machine

- **CPU**
  - Xeon E5-1660
  - June, 2016. 14nm technology
  - 3.2 GHz
  - 8 cores (2x hyperthreaded)
  - 20 MB L3 cache
  - 32 GB DRAM
- **GPU**
  - GeForce GTX 1080
  - May, 2016. 16nm
  - Cuda capability 6.0
  - 8.2 TF Peak (SP)

Performance of Top 500 Machines



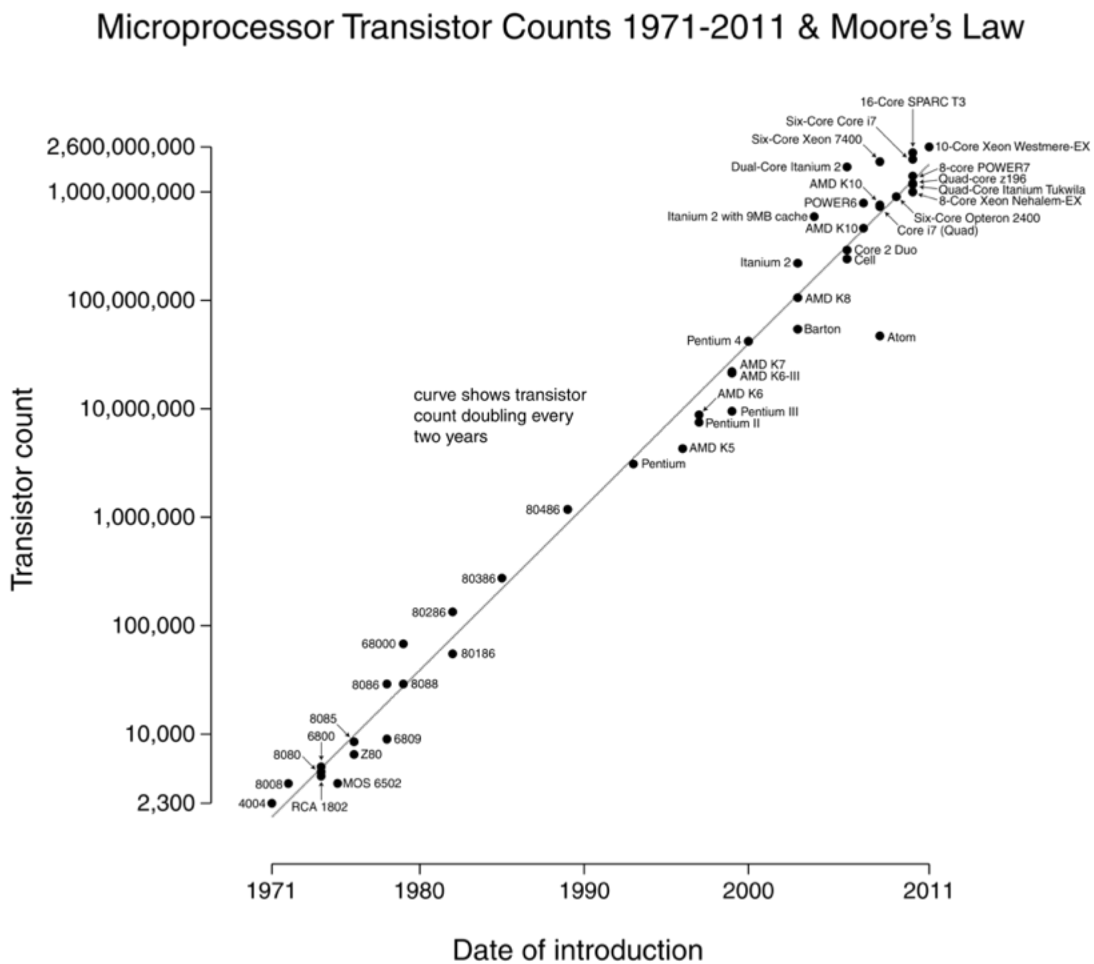
- From presentation by Jack Dongarra
- Machines far off peak when performing HPCG

What Lies Ahead

- **DOE CORAL Program**
  - Announced Nov 2014
  - Delivery in 2018
- **Vendor #1**
  - IBM + nVidia + Mellanox
  - 3400 nodes
  - 10 MW
  - 150 – 300 PF peak
- **Vendor #2**
  - Intel + Cray
  - ~50,000 nodes (Xeon Phi's)
  - 13 MW
  - > 180 PF peak

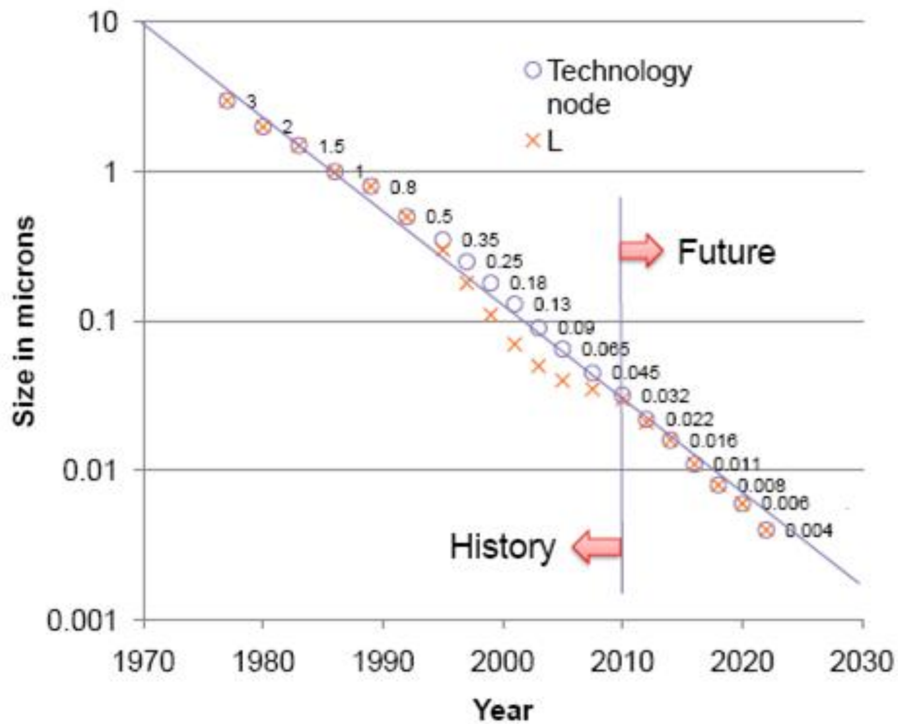
TECHNOLOGY CHALLENGES

Moore's Law



- Basis for ever-increasing computer power
- We've come to expect it will continue

Challenges to Moore's Law: Technical

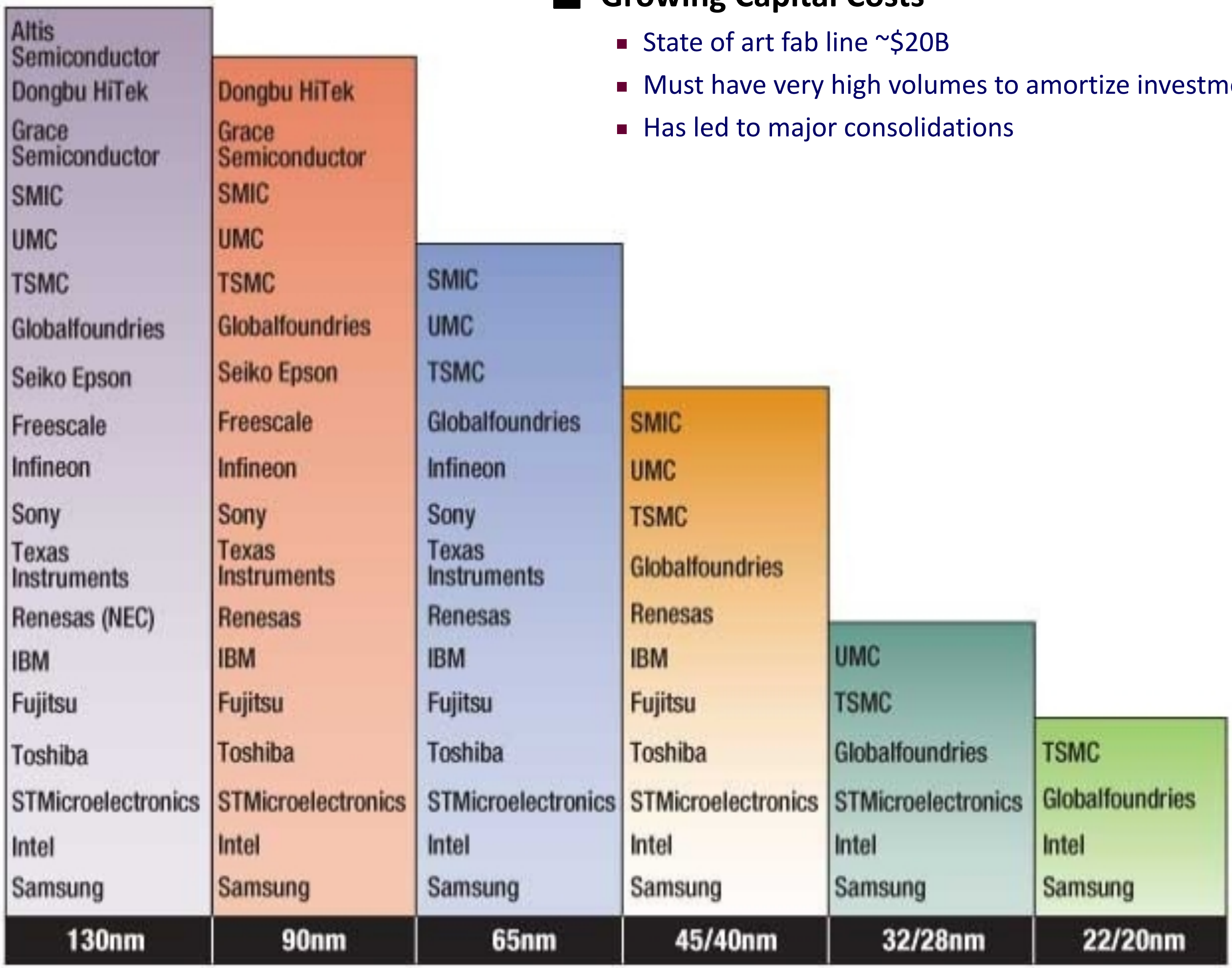


- **2022: transistors with 4nm feature size**
- **Si lattice spacing 0.54nm**

  - Must continue to shrink features sizes
  - Approaching atomic scale

- **Difficulties**
  - Lithography at such small dimensions
  - Statistical variations among devices

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

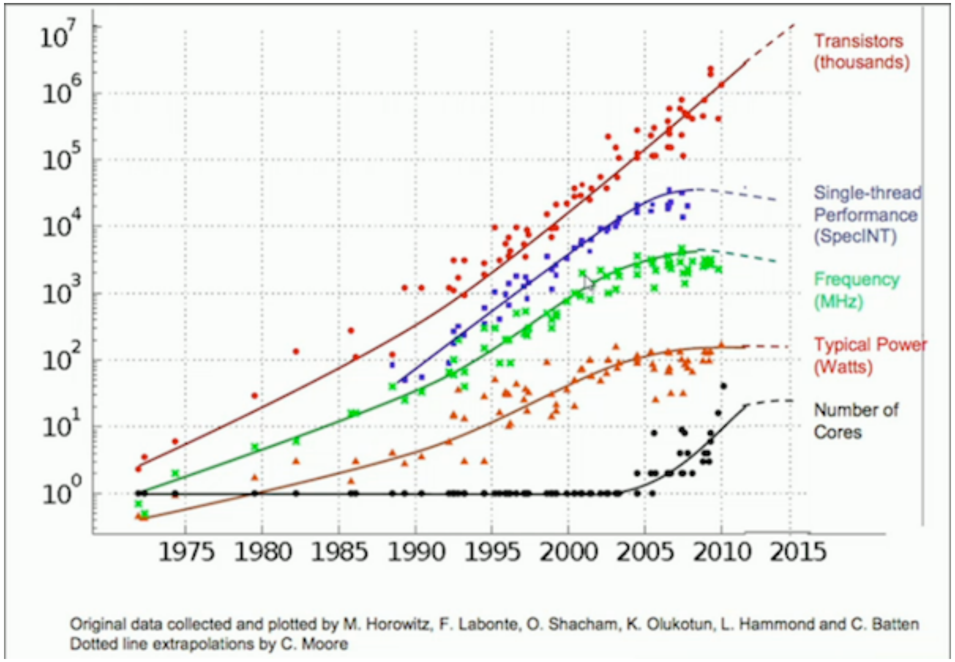
Outcomes

- Devices / chip increase by k^2
- Clock frequency increases by k
- Power / chip constant

Significance

- Increased capacity and performance
- No increase in power

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

Research Challenges

- **Supercomputers**
  - Can they be made more dynamic and adaptive?
    - Requirement for future scalability
  - Can they be made easier to program?
    - Abstract, machine-independent programming models
- **Data-Intensive Computing**
  - Can they be adapted to provide better computational performance?
  - Can they make better use of data locality?
    - Performance & power-limiting factor
- **Technology / Economic**
  - What will we do when Moore's Law comes to an end for CMOS?
  - How can we ensure a stable manufacturing environment?