DS 7347
High-Performance Computing (HPC) and Data Science
Session 3

Robert Kalescky
Adjunct Professor of Data Science
HPC Research Scientist

May 3, 2022

Research and Data Sciences Services
Office of Information Technology
Center for Research Computing
Southern Methodist University

Outline

Session Question

Session Question

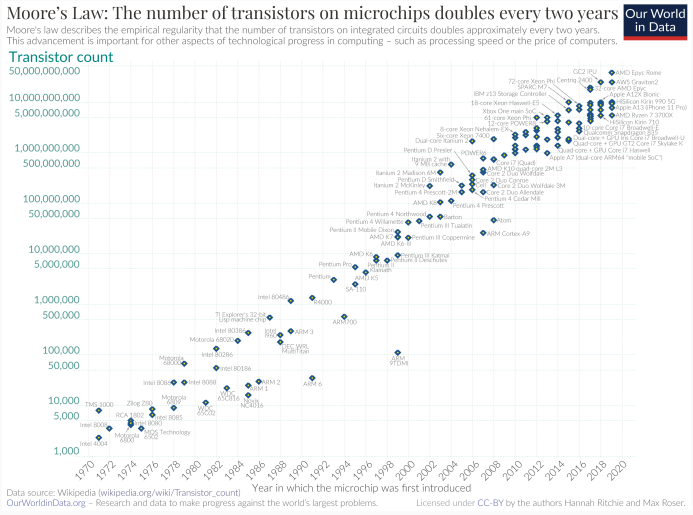
What piece of hardware is the most important to application performance?

Cluster Supercomputers

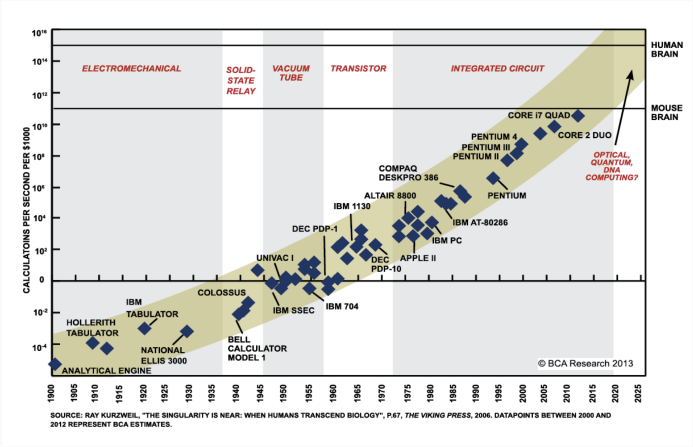
Moore's Law

- Historically, we have depended on hardware advances to enable faster and larger simulations
- In 1965, Gordon Moore observed that the CPU and RAM transistor count about doubled each year
- "Moore's Law" has since been revised to a doubling once every 2 years, with startling accuracy
- Physical limits, e.g. power consumption, heat emission, and even the size of the atom, have currently stopped this expansion on individual processors, with speeds that have leveled off since around 2008

Growth in CPU Transistor Counts



Decrease in Cost Per FLOPS

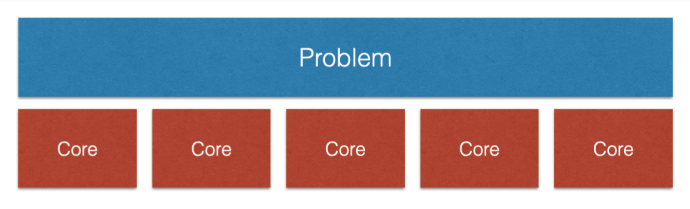


SOURCE: RAY KURZWEIL, "THE SINGULARITY IS NEAR: WHEN HUMANS TRANSCEND BIOLOGY", P.67, *THE VIKING PRESS*, 2006. DATAPOINTS BETWEEN 2000 AND 2012 REPRESENT BCA ESTIMATES.

Additional Problems

- Growing performance discrepancy between processing units and data storage
  - 40% processor performance improvement per year
  - 10% RAM performance improvement per year
- Discrepancy leads to memory-bound programs
  - CPU spends more and more time idle, waiting on data from
- Many simulations require incredible amounts of memory to achieve high-accuracy solutions (PDE, MD, QM, etc.)
  - These cannot fit on a single computer alone

Parallel Computations



- Using multiple processing units (cores, etc.) simultaneously to perform a computation
- Use multiple computers to store data for large problems
- Essentially all modern CPUs have multiple cores

Parallel Hardware

- Levels of parallelism
- Vectorization
- Multicore
- Multi-CPU
- Multi-Node
- Threads versus processes

“I know how to make 4 horses pull a cart - I don’t know how to make 1024 chickens do it.”
Enrico Clementi

What is High-Performance Computing

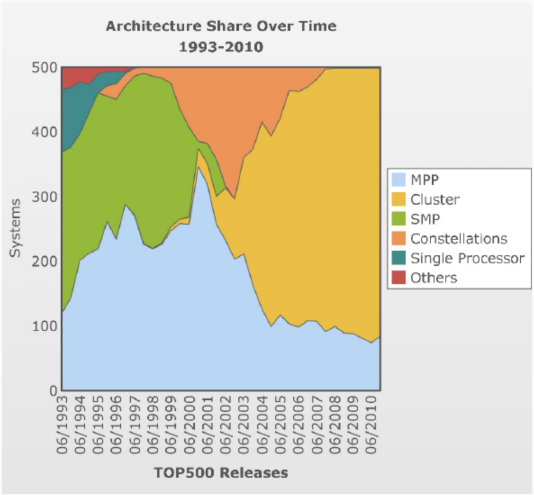
High performance computing (HPC) is the use of computing resources that are significantly more powerful than what is commonly available.

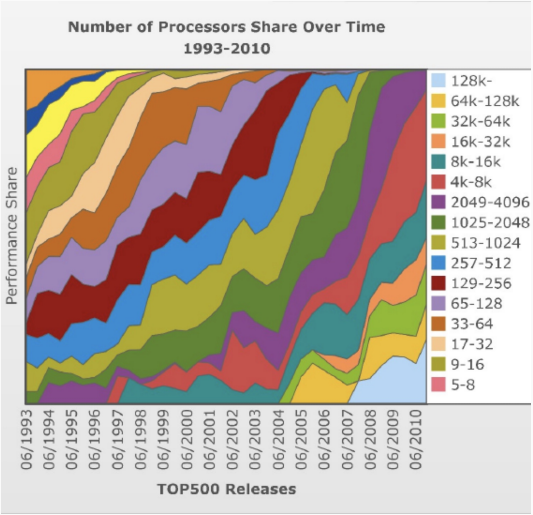
As such, it's always a moving target.

Supercomputer Types

- Single processor
- Shared Memory Parallel (SMP)
- Massively Parallel Processors
- Constellations
- Clusters

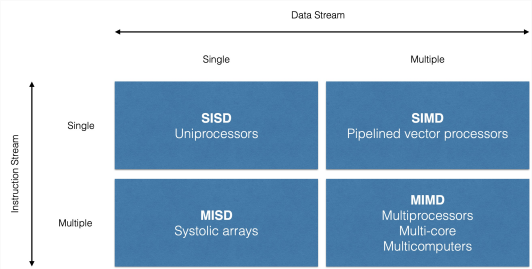
History of Parallel Architectures





Flynn's Parallel Architecture Taxonomy

- Single/multiple instruction streams
  - Number of types of instructions to be performed at once
- Single/multiple data streams
  - Number of data streams to be operated on at once
- Most modern parallel computers are MIMD
- SIMD was popular until 1990s
- MISD never used to large extent



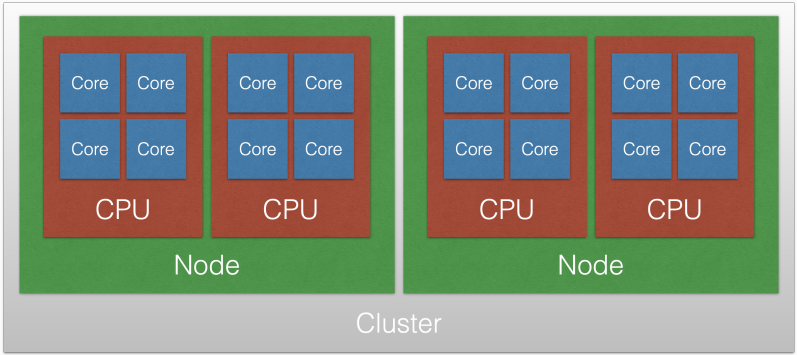
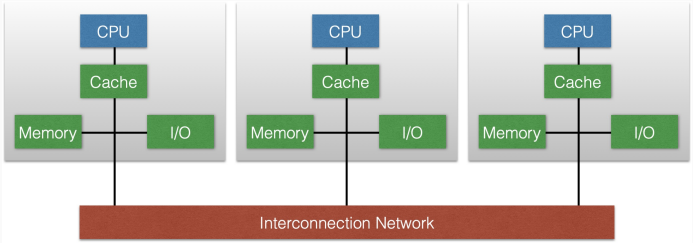


Figure 1: A cluster is a collection of individual computers networked together. Applications can be configured to run on all available compute resources.

Multicomputer Architecture



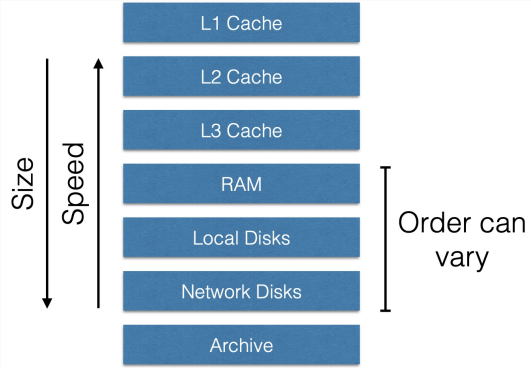
- Each processor only has direct access to its own local memory address space
  - The same address on different processors refers to different memory locations
- Processors interact with one another through passing messages
- Commercial multicomputers typically provide a custom switching network to provide low-latency, high-bandwidth access between processors

Multicomputer Architecture

- Commodity clusters are build using commodity computers and switches/LANs
- Less costly than SMP's
- Increased latency/decreased bandwidth between CPUs
- Theoretically extensible to arbitrary processor counts
- Software becomes complicated
- Networking gets expensive

Cluster Storage Resources

- Usually there is an inverse relationship between storage amount and speed
- Clusters and applications can distort the standard order
  - Aggregation, e.g. RAID
  - Network optimization to a specific topology
  - Applications can be configured and built for specific hardware
  - New hardware technologies



ManeFrame II (M2)

ManeFrame II (M2) Node Types

Type	Quantity	Cores	Memory [GB]	Additional Resources
Standard-Memory	176	36	256	
Medium-Memory-1	35	36	768	
Medium-Memory-2	4	24	768	3 TB SSD local scratch
High-Memory-1	5	36	1,536	
High-Memory-2	6	40	1,536	3 TB SSD local scratch
GPGPU-1	36	36	256	NVIDIA P100 GPU has 3,584 CUDA cores and 16 GB CoWoS
MIC-1	36	64	384	16 GB of high bandwidth (400 GB/s) stacked memory
VDI	5	36	256	NVIDIA Quadro M5000 GPU
v100x8	3	36	768	8 NVIDIA V100 GPUs with 5,120 CUDA cores and 32 GB CoWoS
Faculty Partner Nodes	3			Various research specific NVIDIA GPU configurations
ManeFrame II	354	11,276	120 TB	2.8 PB storage and InfiniBand network

$HOME	• Default file system when logging into M2, e.g. `/users/$USER`. • Space should be used to write, edit, compile programs, and job submission scripts, etc. • Restricted by quotas (200 GB) and backed-up.
$WORK	• Long term storage at `/work/users/$USER`. • Restricted by quotas (8 TB) and not backed-up.
$SCRATCH	• Scratch space at `/scratch/users/$USER`. • Treat $SCRATCH as a volatile file system that is not backed-up.

Unversity Data Center (UDC)

ManeFrame II (M2)

Hot Aisle Containment

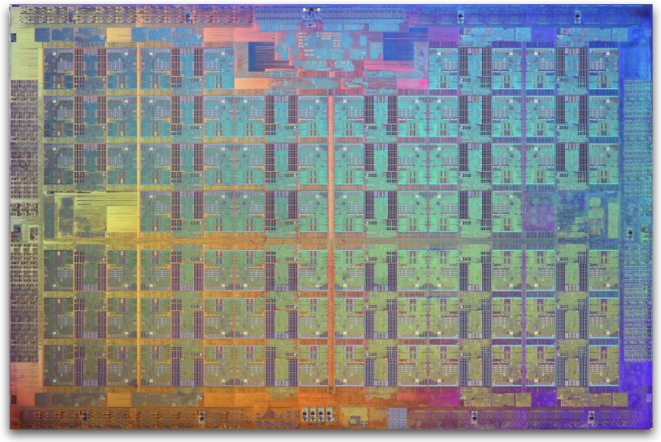
Cooling and Power

Compute Hardware

Intel Xeon Phi Nodes

- Intel Xeon Phi 7230 (also known as “Knights Landing” or “KNL”) processors
- 64 1.30 GHz cores based on the Atom “Silvermont” architecture
- Cache: 32 MB L2
- Cache: 16 GB of high bandwidth (400 GB/s) stacked memory
- Memory: DDR4 2400 MHz, 115.2 GB/s
- Vector Extensions: AVX-512 (512 bit width)
- Hardware-based support for up to four concurrent threads

Intel Xeon Phi Nodes



Intel Xeon “Broadwell” Nodes

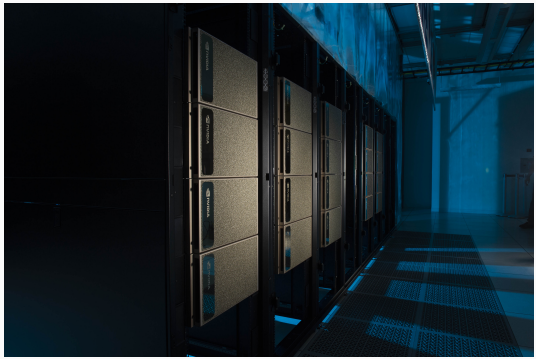
- Dual Intel Xeon E5-2695v4 2.1 GHz 18-core “Broadwell” processors
- Cache: 45 MB L3
- Memory: DDR4 2400 MHz, 76.8 GB/s
- Vector Extensions: AVX2 (256 bit width)

Intel Xeon “Skylake” Nodes

- Dual Intel Xeon Gold 6154 3.0 GHz 18-core “Skylake” processors
- Cache: 24.75 MB L3
- Memory: DDR4 2666 MHz, 119.21 GB/s
- Vector Extensions: AVX-512 (512 bit width)

NVIDIA SuperPOD DGX Nodes

- Dual AMD Epyc 7742 2.25 GHz 64-core “Rome” processors
- Cache: 256 MB L3
- Memory: DDR4 3200 MHz, 204.8 GB/s
- Vector Extensions: AVX2 (256 bit width)



Vector Extensions

Array Programming

- Operations are exclusively done over vectors, matrices, and tensors instead of scalars
- Scalars in array programming are vectors with a single index
- Array programming simplifies programming linear algebra algorithms
- Array programming languages include MATLAB, Octave, R, Julia, and Python via NumPy

```
>>> from numpy import matrix, linalg
>>> A = matrix([[1,2,3],[4,5,6],[7,8,9]])
>>> x = matrix([[1],[2],[3]])
>>> print(A.T)
[[1 4 7]
 [2 5 8]
 [3 6 9]]
>>> print(A*x)
[[14]
 [32]
 [50]]
```

Single Instruction, Multiple Data

- A single operation is applied to multiple data simultaneously
- Allows some linear alegra operations to be done via vectors instead of single elements
- Modern systems are MIMD where each processor or core can perform independent SIMD operations
- SIMD instructions are CPU architecture specific, *i.e.* not all CPU's have the same SIMD instruction sets

Vectorization Example

```
1  for (int i=0; i<16; ++i) {
2     C[i] = A[i] + B[i];
3  }
4
5  for (int i=0; i<16; i+=4) {
6     addFourThingsAtOnceAndStoreResult(&C[i], &A[i], &B[i]);
7  }
```

Advanced Vector Extensions

- Advanced Vector Extensions (AVX) are a set of SIMD extensions to the x86 instruction set
- Initially developed by Intel in 2008
- Implemented in Intel and AMD CPU's since 2011

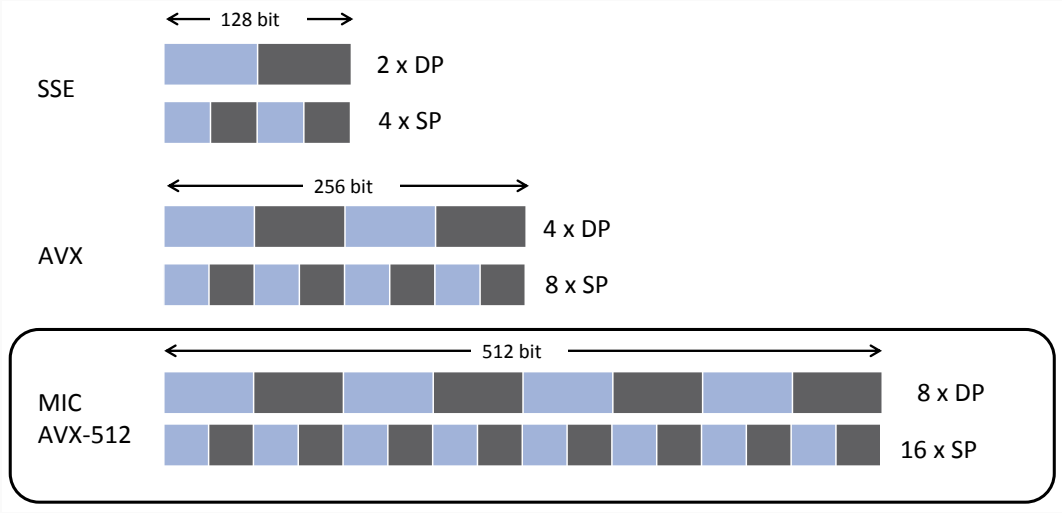
AVX Versions

AVX
- Expanded many of the earlier SSE floating point instructions from 128 to 256 bits
- Several new instructions
- Intel Sandy Bridge and AMD Bulldozer and newer architectures

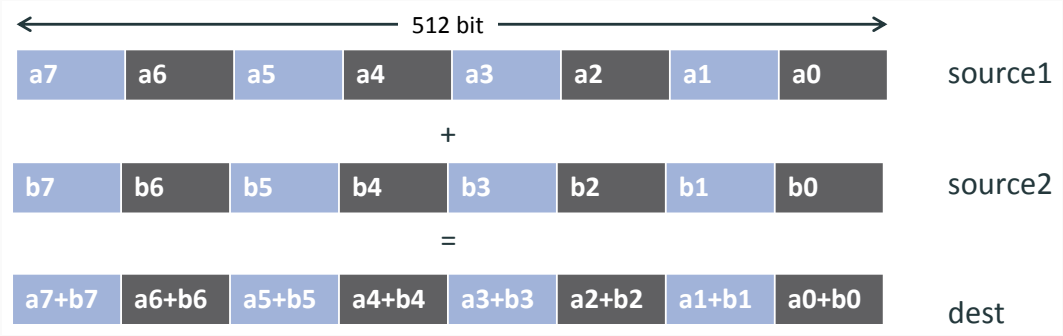
AVX2
- Expanded many of the earlier SSE integer instruction from 128 to 256 bits
- Several new instructions
- Intel Haswell and AMD Carrizo and newer architectures

AVX-512
- Expanded AVX instructions from 256 to 512 bits
- Several new instructions
- Intel Xeon Phi x200 series CPU's

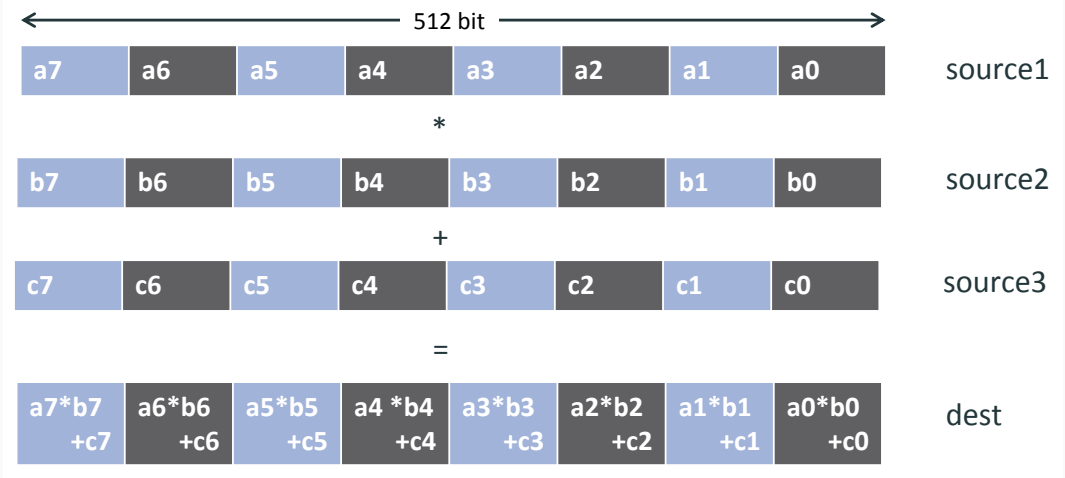
Vector Extension Widths



Vector Extension Addition



Vector Extension Fused Multiply-Add



Readings and Assignments

Readings and Assignments

Readings

- Eijkhout sections [1.5–1.6]

Assignment

No assignments.