Introduction to High-Performance Computing

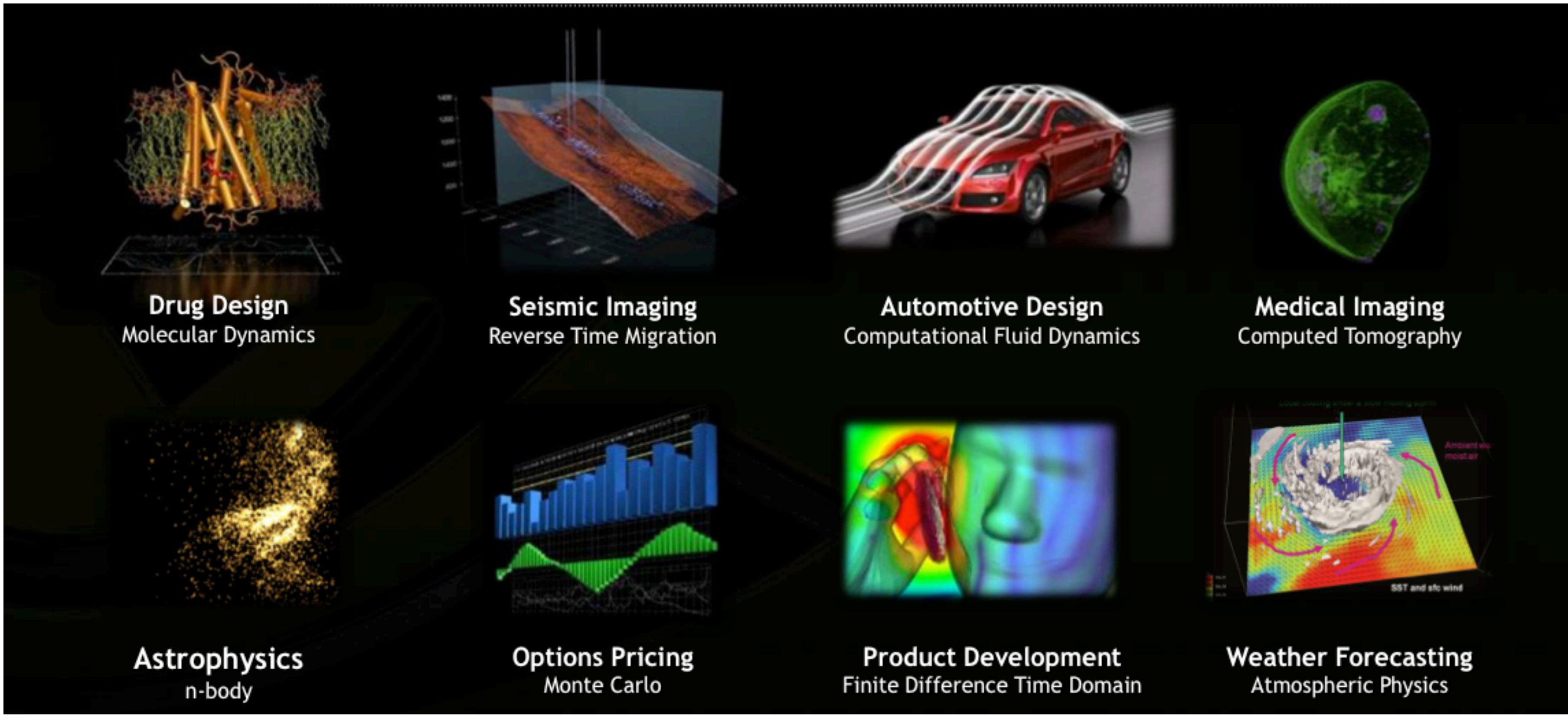
Lecture’s objectives

- Explain different types of “big problem”
- Compare and contrast different types of parallel computer architecture
- Identify the difference between node, socket, and core

Scientific discovery

- Historically, people said science stands on two legs:
  - Experiment
  - Theory
- Then, science got the third leg:
  - Computing
- Now, science has the fourth leg:
  - Big data

Computing: the 3rd pillar



Computing and *Big* problems

- Computing-based science discovery deals with Big problems

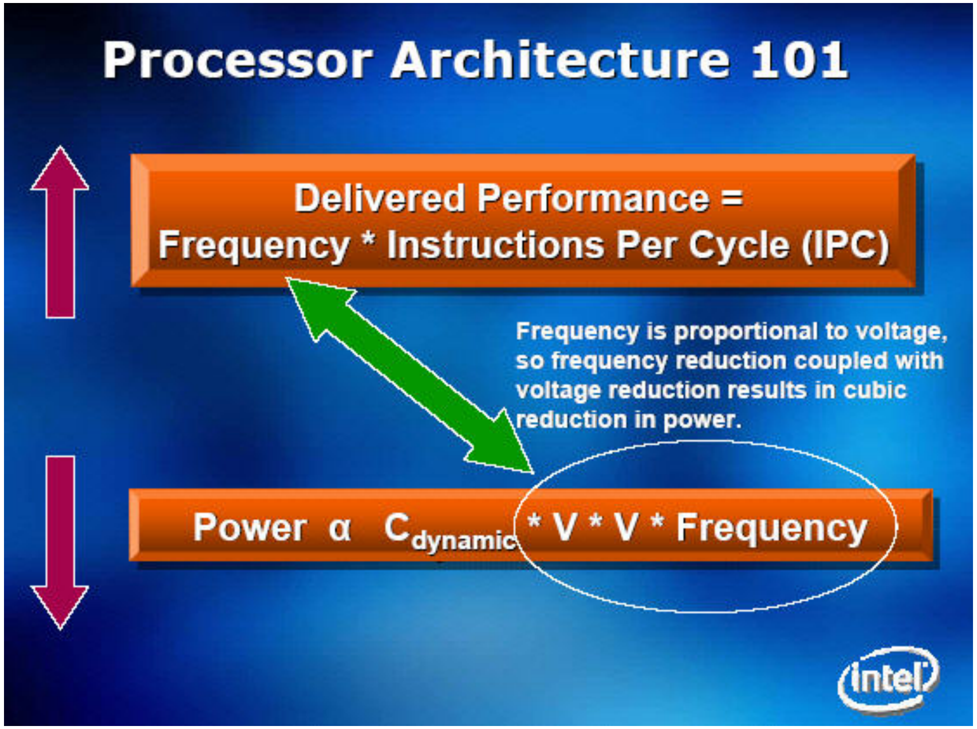
- For example:
  - It will take many days to perform a very short Molecular Dynamic (MD) simulation of 20,000 atoms
  - Whole genomic analysis of 100 human genomes using currently available desktop computers could take 30+ years

Types of *Big* problem

- Compute Intensive
  - A single problem requires a large amount of computation
- Memory Intensive
  - A single problem requires a large amount of memory
- Data Intensive
  - A single problem operates on a large amount of data
- High Throughput
  - Many copies of the same problem to be executed in parallel

Big problems require *Big* computers

- Faster CPUs
  - Processor clock speeds have flattened out
  - The fastest commercially available chip is about 5.0 GHz
  - Clock speed is limited by power consumption, heat dissipation, current leakage



Big problems require *Big* computers

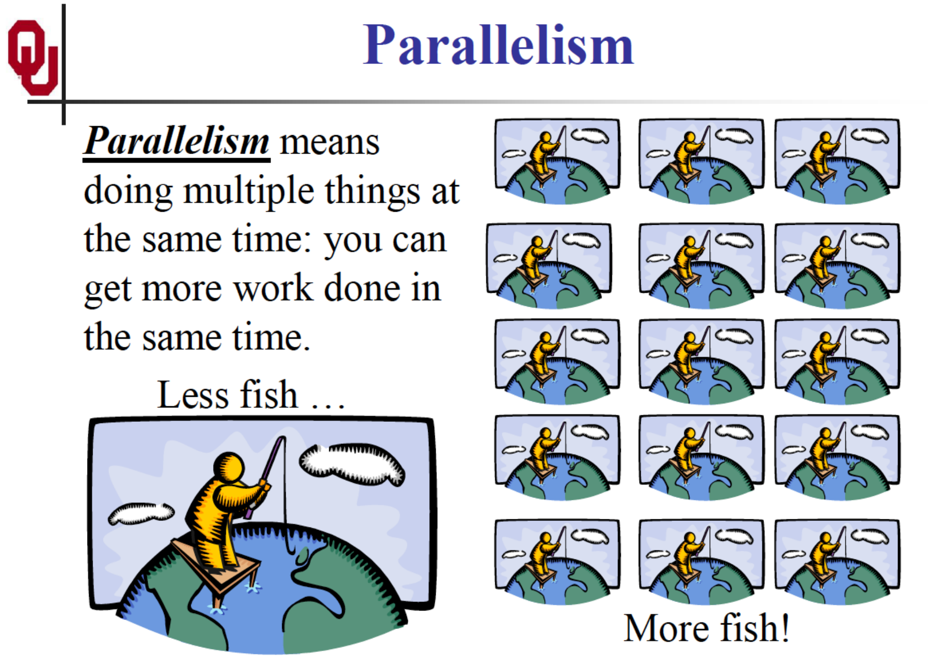
- Many CPUs
- Lots of memory
  - Limited by how much memory a CPU can support
- Large amount of storage space

Parallelism, the path toward future performance gains

- Trend is toward multi-core and many core

What is HPC?

- High Performance Computing is the "practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop/laptop computer"[1]



[1]http://insidehpc.com/hpc-basic-training/what-is-hpc/
Picture from "supercomputing in plain English", http://www.oscer.ou.edu/education.php

Parallel Computer Architectures

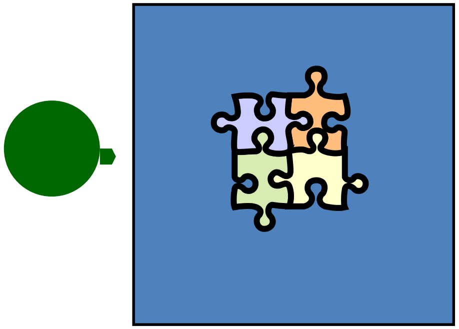
The Jigsaw Puzzle

- Say you have a jigsaw puzzle with 1000 pieces
- How can you put it together as fast as possible?

Adapted from “supercomputing in plain English”, http://www.oscer.ou.edu/education.php

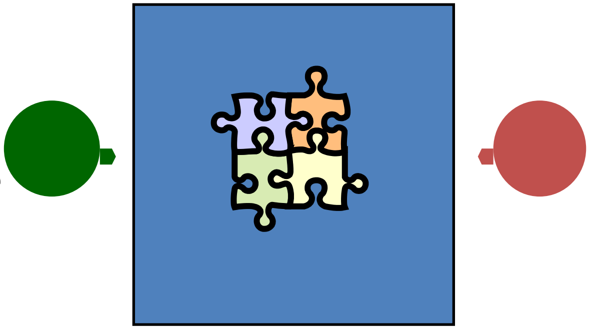
The Jigsaw Puzzle

- Serial Computing
  - You sit down at the table by yourself and put together all 1000 pieces, one after the next
  - It takes you 1 hour to assemble the puzzle
  - Well done!
  - But can we do better?



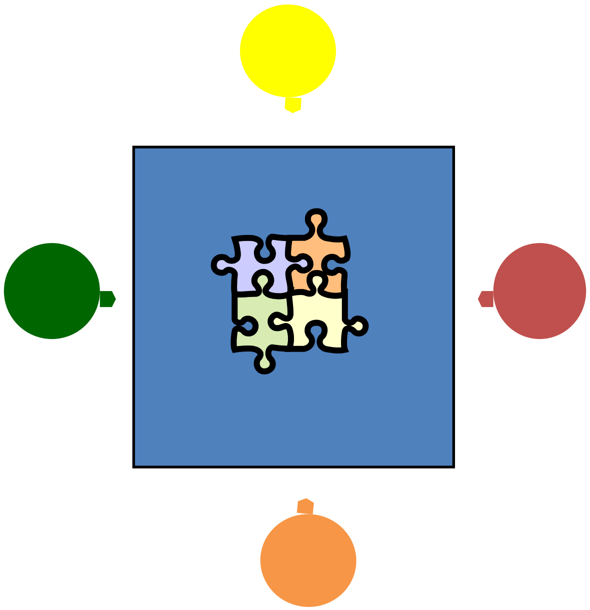
The Jigsaw Puzzle

- Shared Memory Parallelism
  - If your friend sits across from you, then he can work on his half of the puzzle and you can work on yours
  - Once in a while, you'll both reach into the pile for the same piece (you will **contend** for the same resource) which causes a little slowdown
  - From time to time, you'll have to work together (**communicate**) at the interface of your halves
  - If all goes well, you'll get a 2x speedup
  - But we have communication and contention overhead so it will probably take more like 35 minutes instead of 1 hour



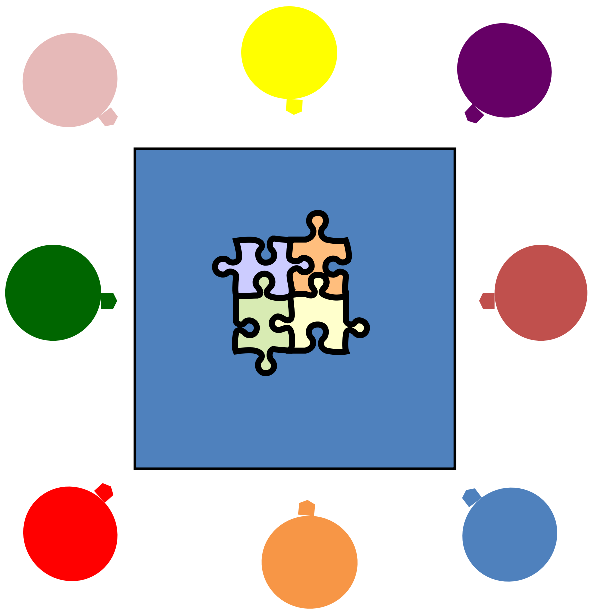
The Jigsaw Puzzle

- More Shared Memory Parallelism?
  - Let's say you add two more friends
  - Each person works on their quadrant of the puzzle, we should get a 4x speedup, right?
  - But, there will be a lot more contention for pieces and a lot more communication
  - Instead of putting together puzzle in ¼ hours or 15 minutes, you'll probably be closer to 20 minutes



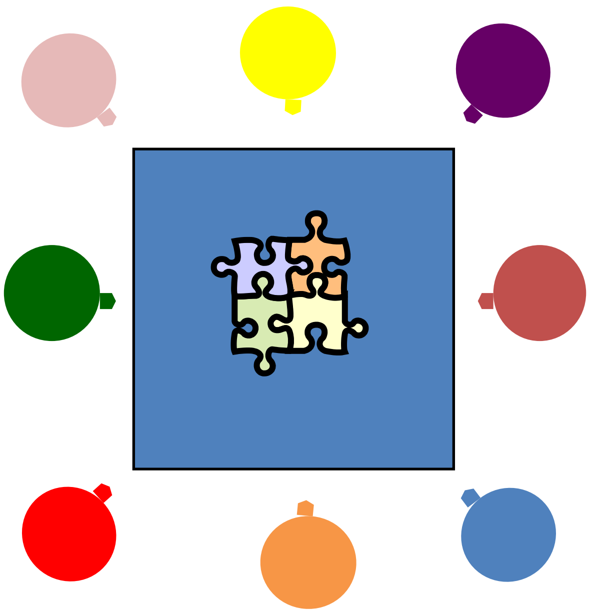
The Jigsaw Puzzle

- Diminishing returns?
  - Let’s say you add 4 more friends
  - MUCH more contention for pieces and a TON more communication
  - If you actually manage to get a 8x speedup it would be very impressive



The Jigsaw Puzzle

- But...
  - Let’s assume the 8 of you are extremely good at working together
  - Maybe you can get an almost 8x speedup
  - But we want to go faster yet
  - Problem! There’s only enough space for 8 people at the table!

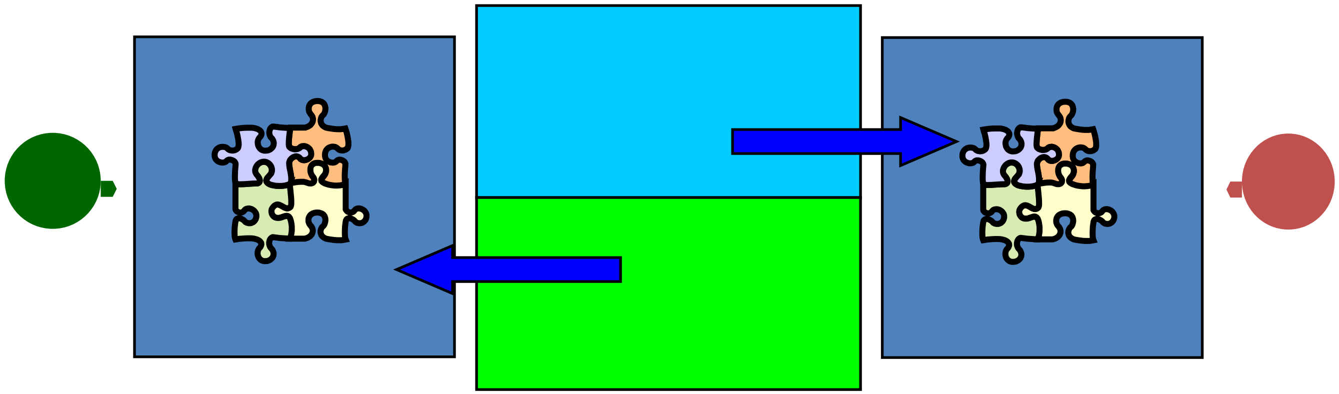


The Jigsaw Puzzle

- Distributed Parallelism
  - You sit at table 1 and your friend sits at table 2
  - PRO: Plenty of elbow room
  - PRO: You can work without contention for resources
  - CON: Communication is much more difficult. You need to carry pieces from one table to the other to assemble them
  - Other problems?

The Jigsaw Puzzle

- Load balancing and Domain Decomposition
  - Say the puzzle is half blue sky and half green grass
  - Put all the blue pieces on one table and all the green pieces on the other
  - Pieces/table are roughly equal so each half should be assembled in roughly equal amount of time



A word about load balancing

- **Q:** How long does it take to finish 100 parallel calculations that all start at the same time?

A word about load balancing

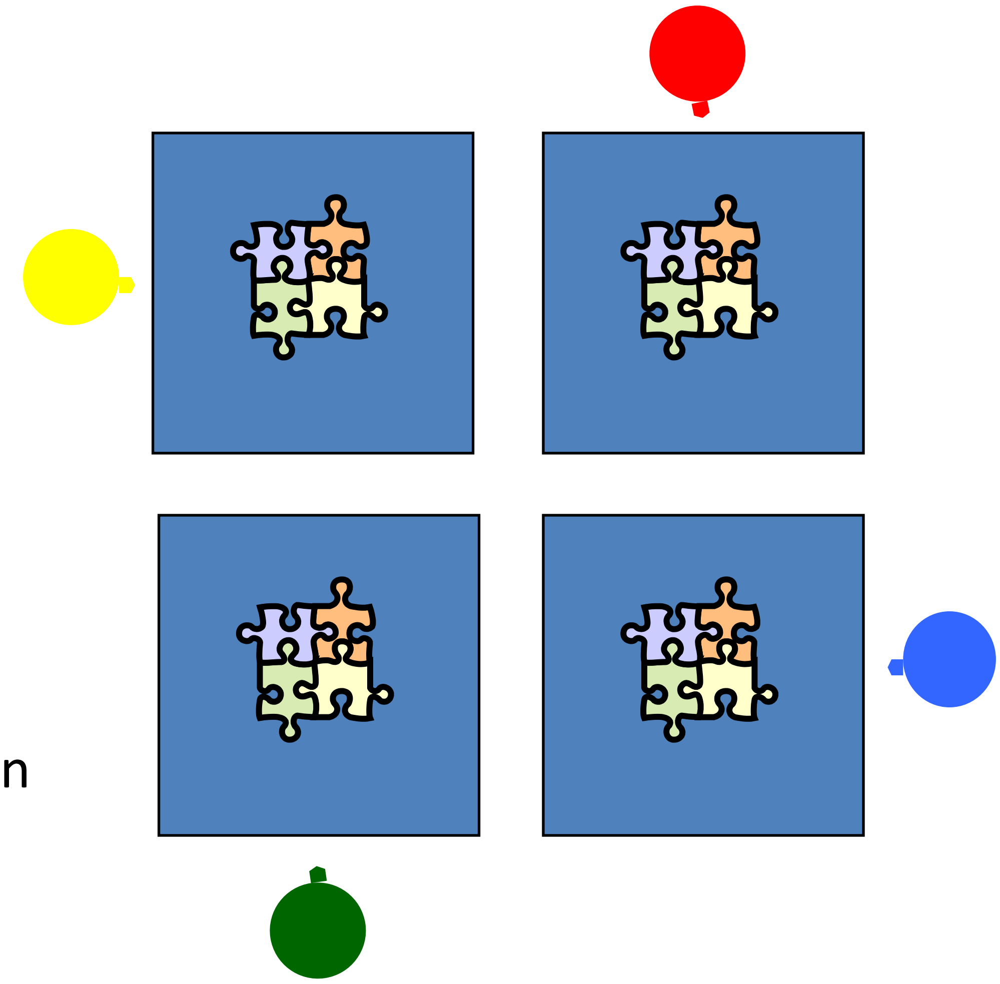
- **Q:** How long does it take to finish 100 parallel calculations that all start at the same time?

- **A:** However long the slowest of the 100 calculations takes

If one calculation is significantly slower than all the rest, parallelism will not help you

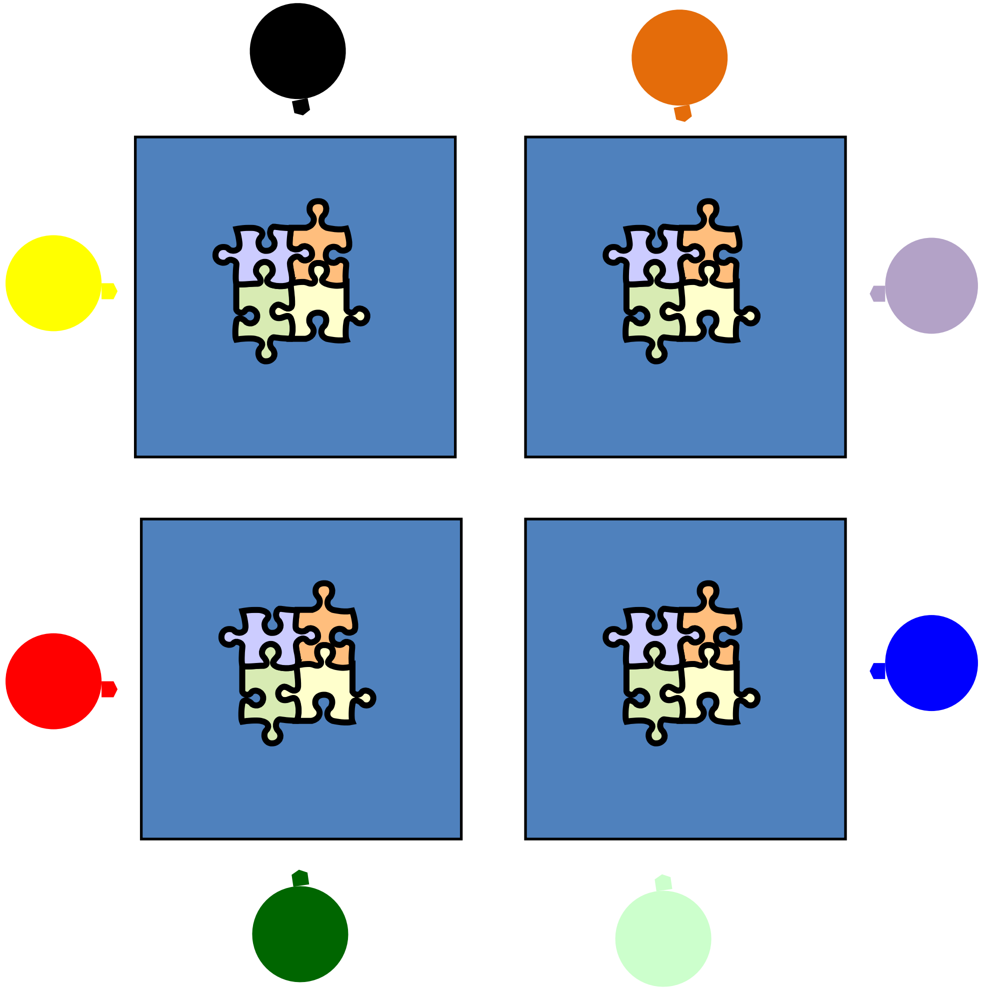
The Jigsaw Puzzle

- More Distributed Parallelism?
  - It's very easy to keep adding more tables
  - But…
    - Communication
    - Load balancing
    - Domain decomposition



The Jigsaw Puzzle

- Hybrid Parallelism
  - Combine distributed and shared models
  - What are the problems?
  - What about advantages?



Some definitions

- **Core**: smallest computation unit that can run a program (used to be called a processor, still is, also called a CPU — Central Processing Unit)

- **Socket**: a computational unit, packaged as one and usually made of a single chip often called processor. Modern sockets carry many cores (2, 4 on most laptops, 8 to 16 on most servers)

- **Node**: a stand-alone computer system that contains one or more sockets, memory, storage, etc. connected to other nodes via a fast network interconnect

Question

- Find out how many cores, sockets, how much main memory (aka RAM), and hard drive your laptop has.

A super computer is...

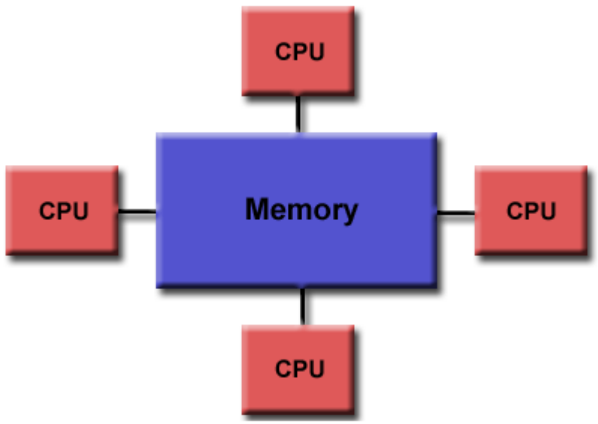
- A collection of small computers, called **nodes**, hooked together by an **interconnection network** (or interconnect for short)

- Software that allows nodes to communicate with one another

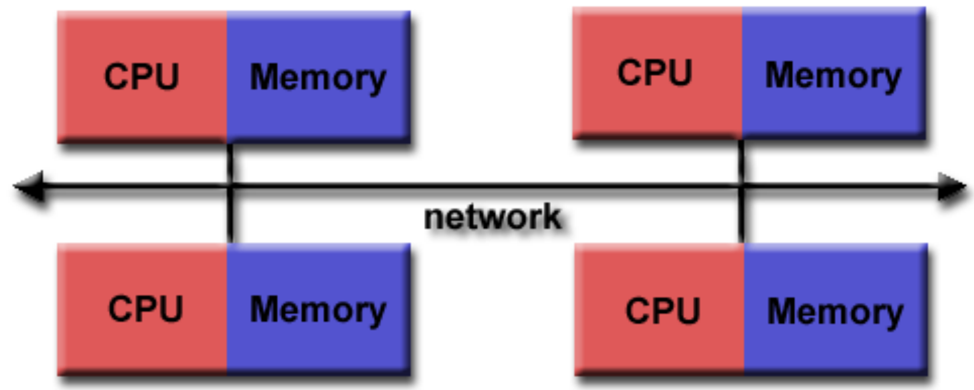
- All of these nodes work together as if they are one big computer … a supercomputer!

Parallel computer architectures

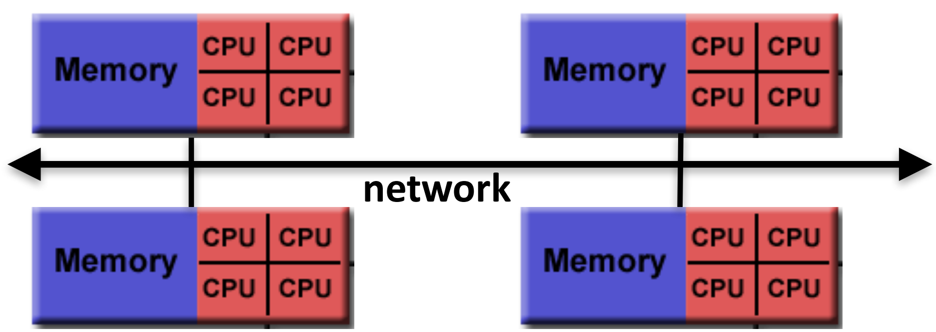
Shared Memory



Distributed Memory



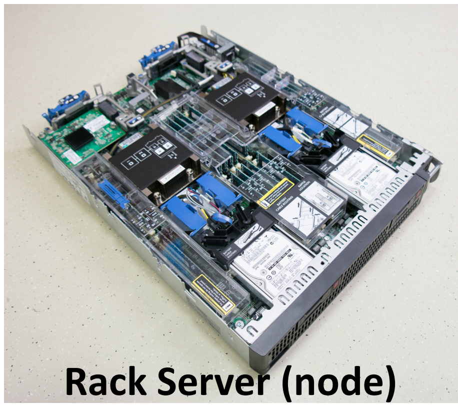
Hybrid



Images from https://computing.llnl.gov/tutorials/parallel_comp/

What is HPC?

From node to cluster



Rack Server (node)

Cabinet

Cluster

Data Center

Fastest super computers

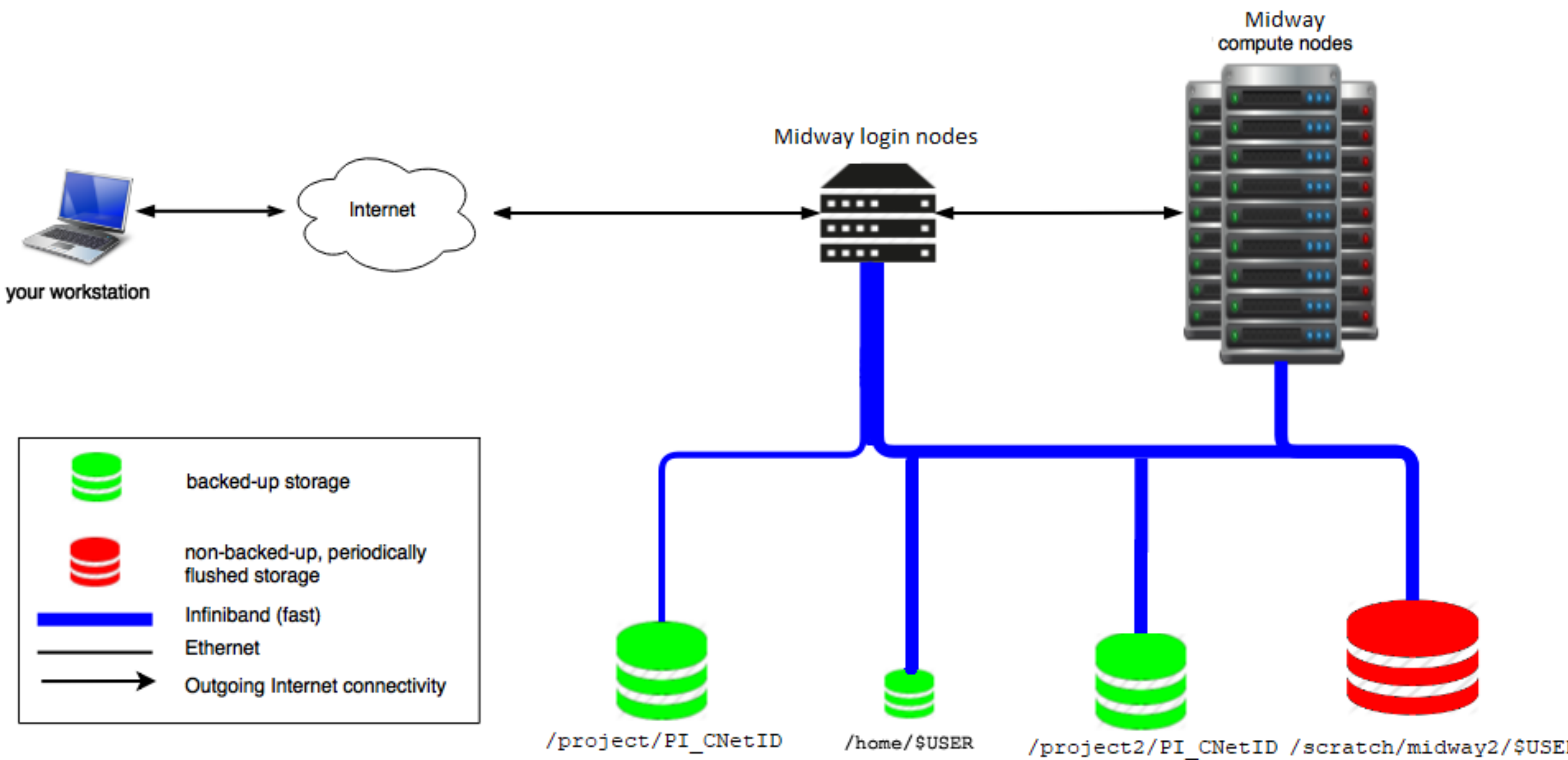
1. Summit - Oak Ridge National Laboratory
   – Cores: 2,414,592 Pflop/sec: 148.60 Power: 10.096 MW
2. Sierra Lawrence Livermore National Lab
   – Cores: 1,572,480 Pflop/sec: 94.64 Power: 7.438 MW
3. Sunway TaihuLight National Supercomputing Center in Wuxi, China
   – Cores: 10,649,600 Pflop/sec: 93.01 Power: 15.371 MW

1 Pflop = 1,000,000,000,000,000 floating point operations

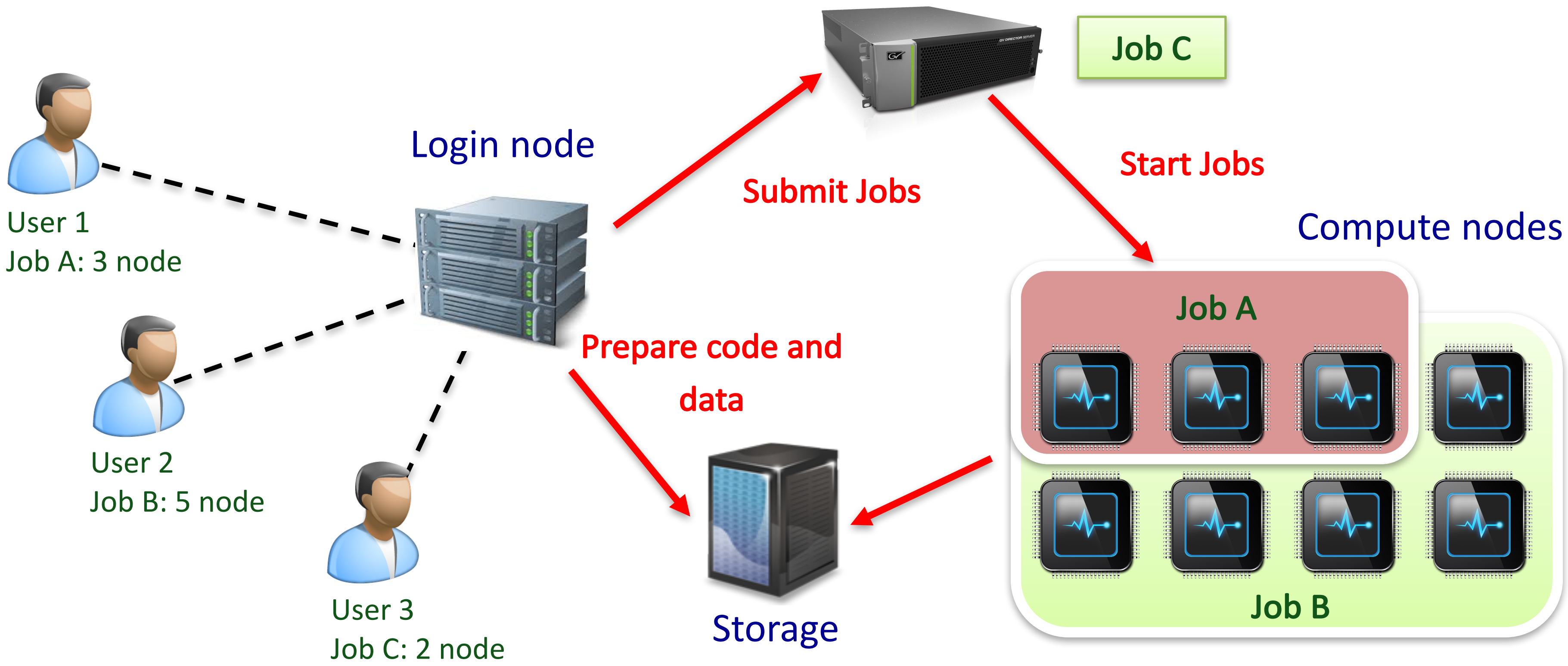
The RCC's Midway system

- Compute
  - Login nodes
  - Compute nodes
    - 300+ Tightly coupled nodes
    - 4 Large memory nodes
    - 6 GPU nodes
- Storage
  - home, project, and scratch spaces
- Software
  - Commercial as well as open source

Schematic of a typical cluster



Job scheduling on Midway



Takeaways

- Computing is the third pillar of scientific discovery
- Computing-based scientific discovery deals with Big problems
- A parallel computer (aka supercomputer) is a collection of small computers working together
- Shared memory, distributed memory, and hybrid are types of parallel computer architectures

References

- The following resources were used in preparation of this presentation:
  - D. Eadline, High Performance Computing for Dummies, Second AMD Special Edition, Wiley Publishing Inc., 2011.
  - S. Rankin, An Introduction to High Performance Computing, https://www.hpc.cam.ac.uk/files/introduction_to_hpc-nov2018-handout.pdf
  - S. Lantz, Effective Use of High Performance Computing, http://www.cac.cornell.edu/slantz/CIS4205/, 2009.
  - G. Hager and G. Wellein, Introduction to High Performance Computing for Scientists and Engineers, CRC Press, 2010.