OUTLINES

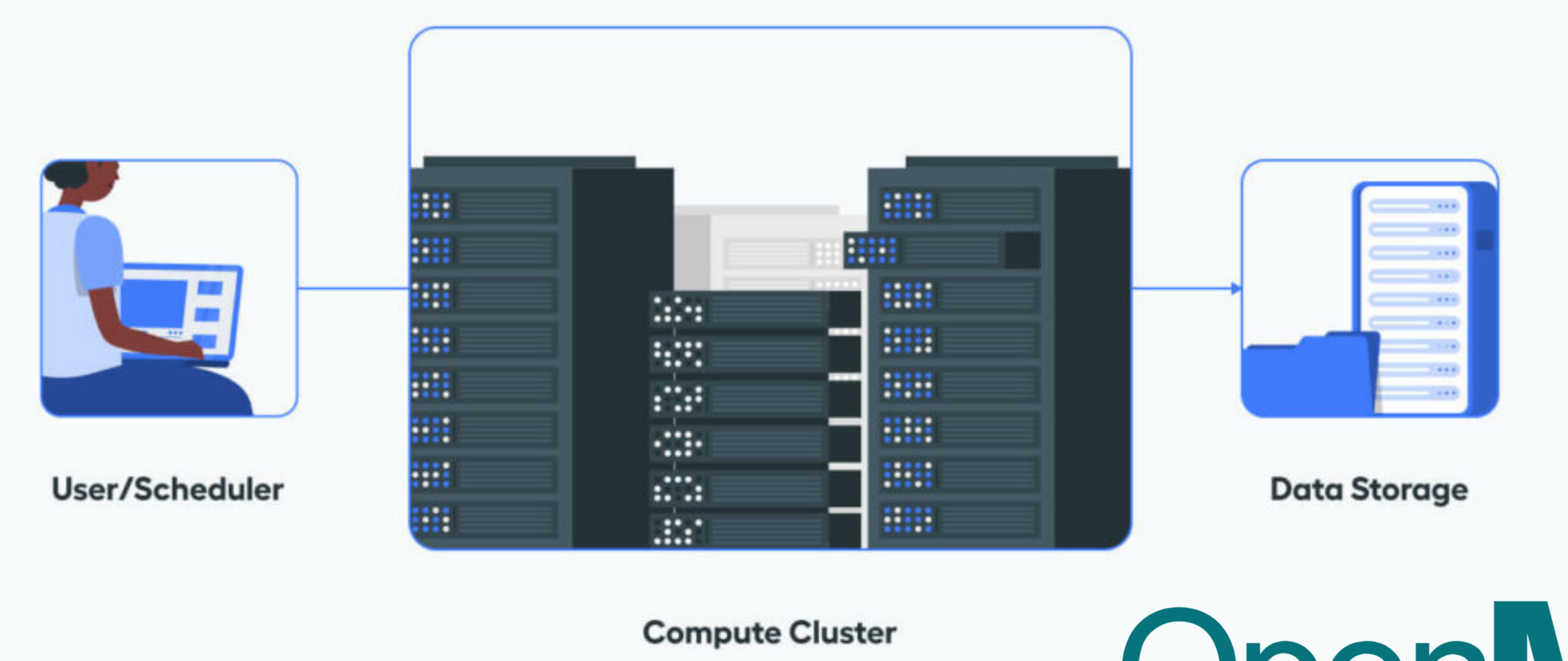
Parallel Programming(CDSC 604)

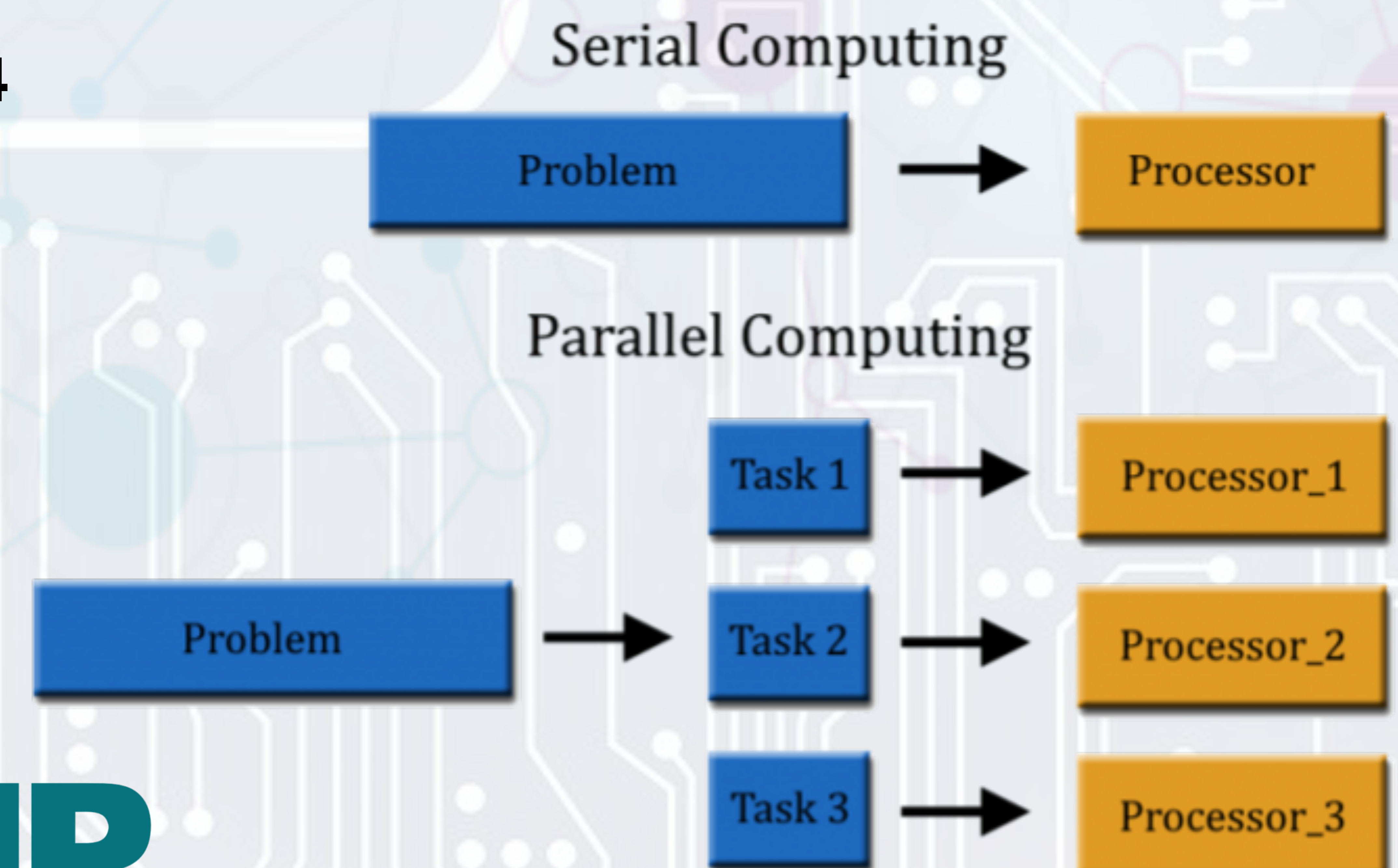
Mesfin Diro Chaka

Computational Data Science Program

Addis Ababa University

March, 2024





Why use Parallel computing?

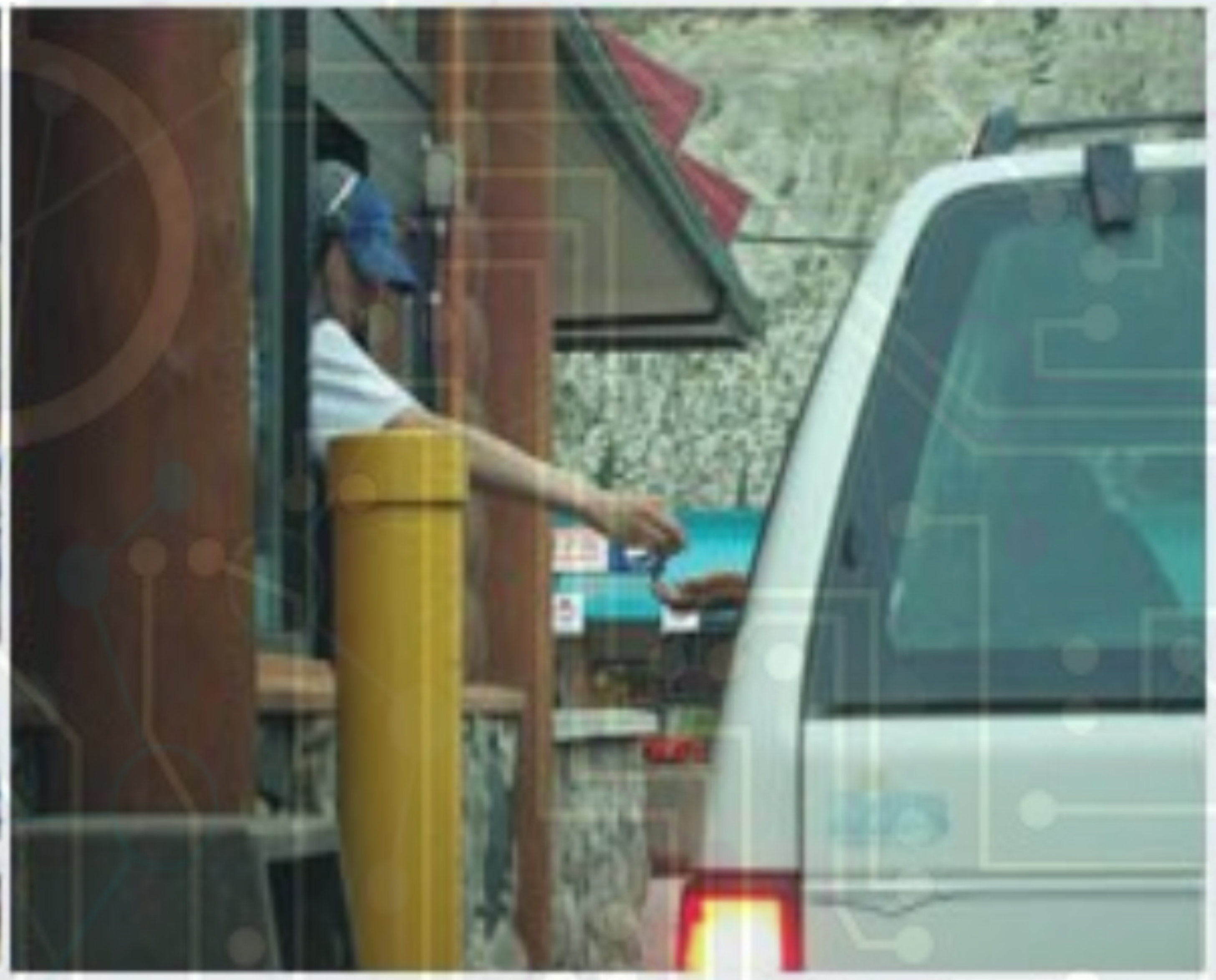
Part 1: Lecture 2

Why use Parallel computing?

- **The Real World is Massively Parallel**
- **In the natural world, many complex, interrelated events are happening at the same time, yet within a temporal sequence.**

Auto Assembly

Jet Construction



Drive-thru Lunch

Why use Parallel computing?

Why use Parallel computing?

- **The Real World is Massively Parallel**
- **In the natural world, many complex, interrelated events are happening at the same time, yet within a temporal sequence.**

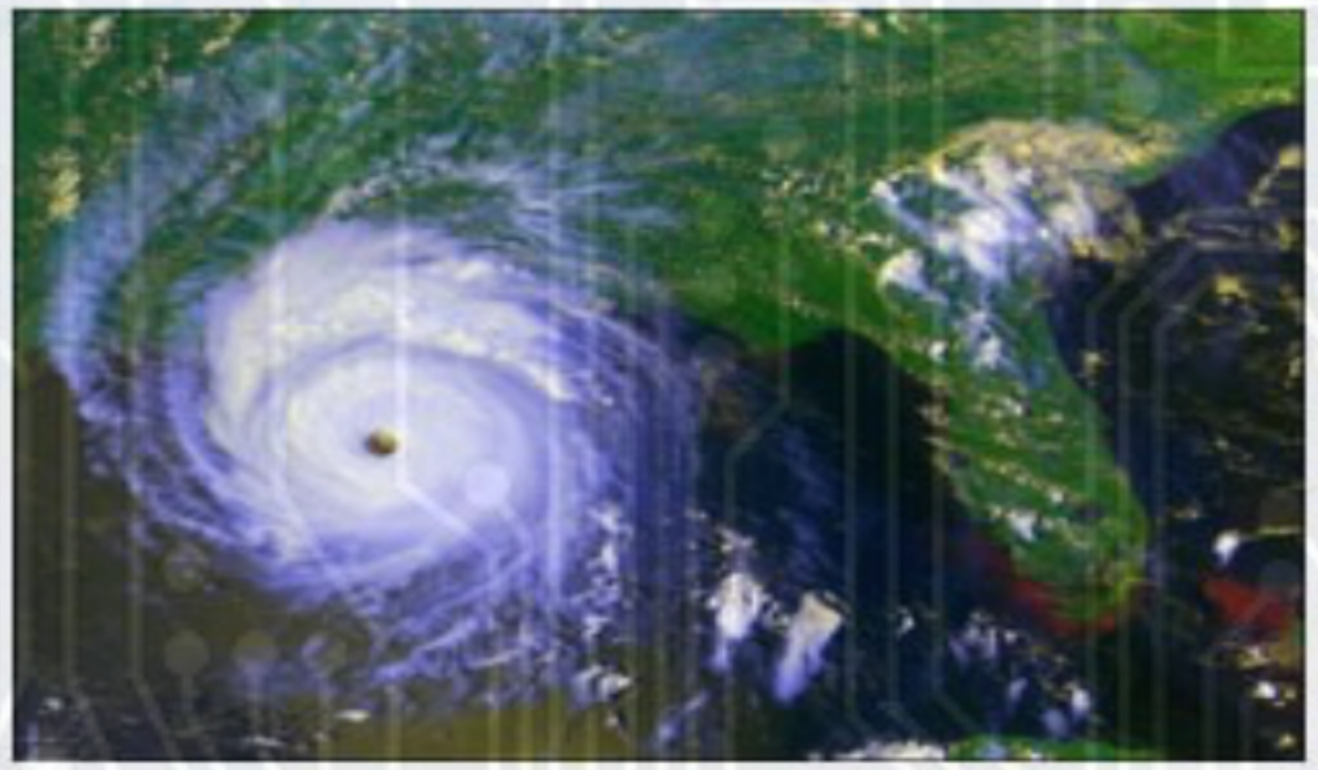
Rush Hour Traffic

Plate Tectonics

Weather

Galaxy Formation

Planetary Movments



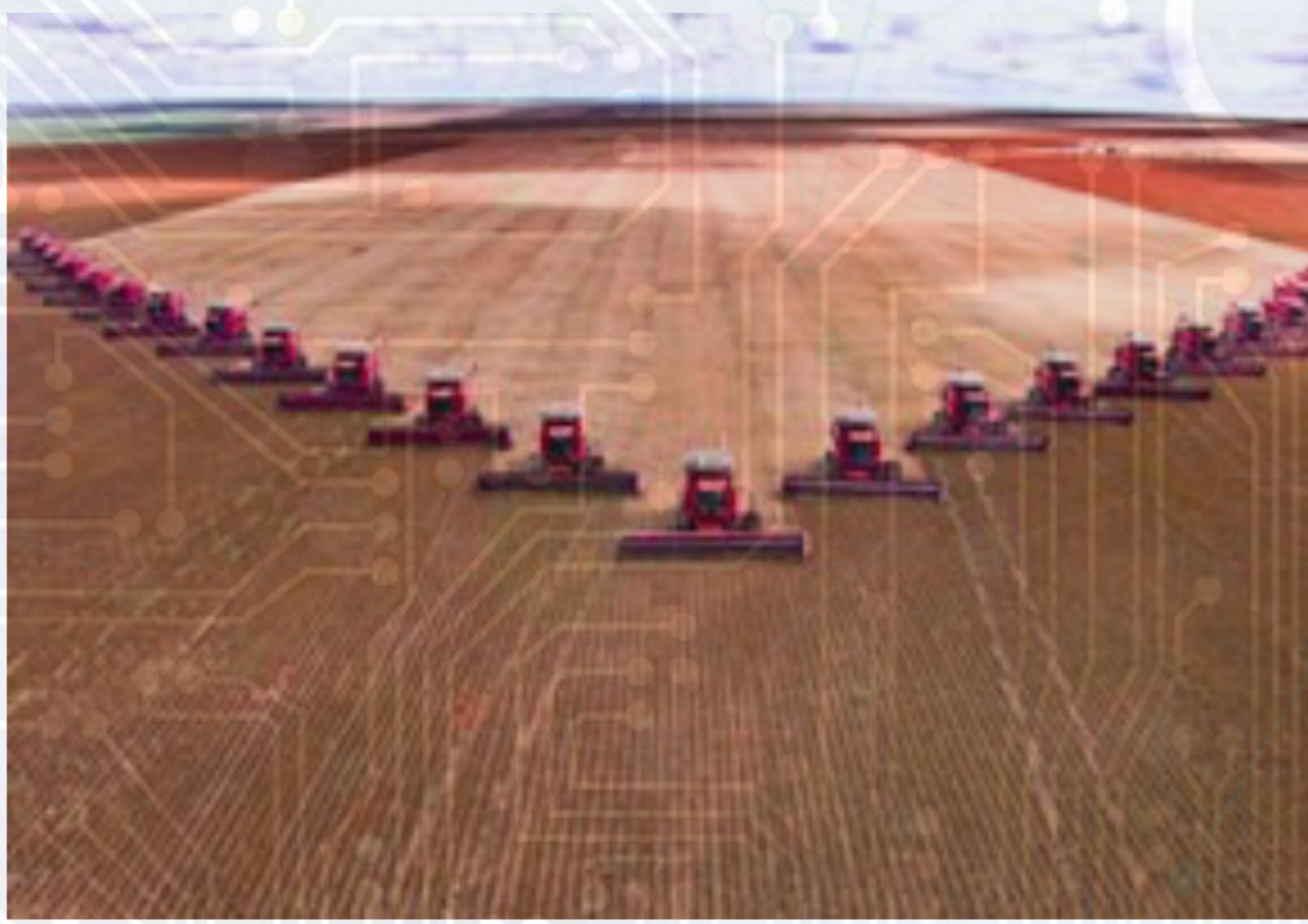
Climate Change

Why use Parallel computing?

Part 1: Lecture 2

Benefits of using Parallel computing

- **SAVE TIME AND/OR MONEY**
  - **In theory, throwing more resources at a task will shorten its time to completion, with potential cost savings.**
  - **Parallel computers can be built from cheap, commodity components.**



Why use Parallel computing?

Part 1: Lecture 2

Benefits of using Parallel computing

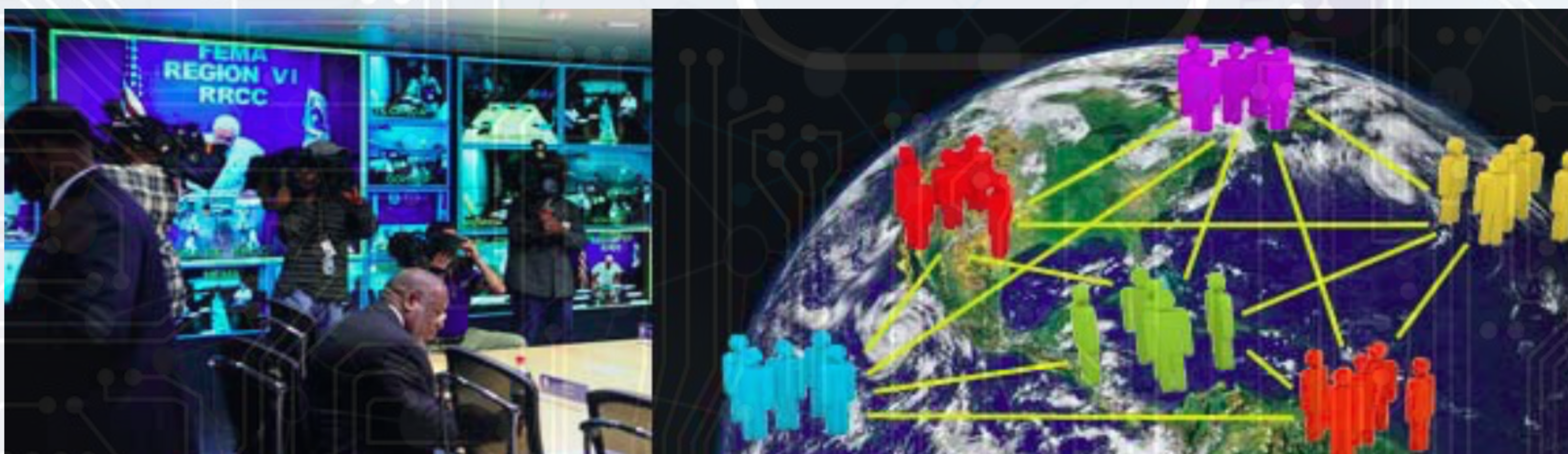
- **SOLVE LARGER / MORE COMPLEX PROBLEMS**
  - Many problems are so large and/or complex that it is impractical or impossible to solve them using a serial program, especially given limited computer memory.
  - Example: "Grand Challenge Problems" (en.wikipedia.org/wiki/ Grand_Challenge) requiring petaflops and petabytes of computing resources.
  - Example: Web search engines/databases processing millions of transactions every second

Why use Parallel computing?

Part 1: Lecture 2

Benefits of using Parallel computing

- **PROVIDE CONCURRENCY**
  - **A single compute resource can only do one thing at a time. Multiple compute resources can do many things simultaneously.**
  - **Example: Collaborative Networks provide a global venue where people from around the world can meet and conduct work "virtually"**

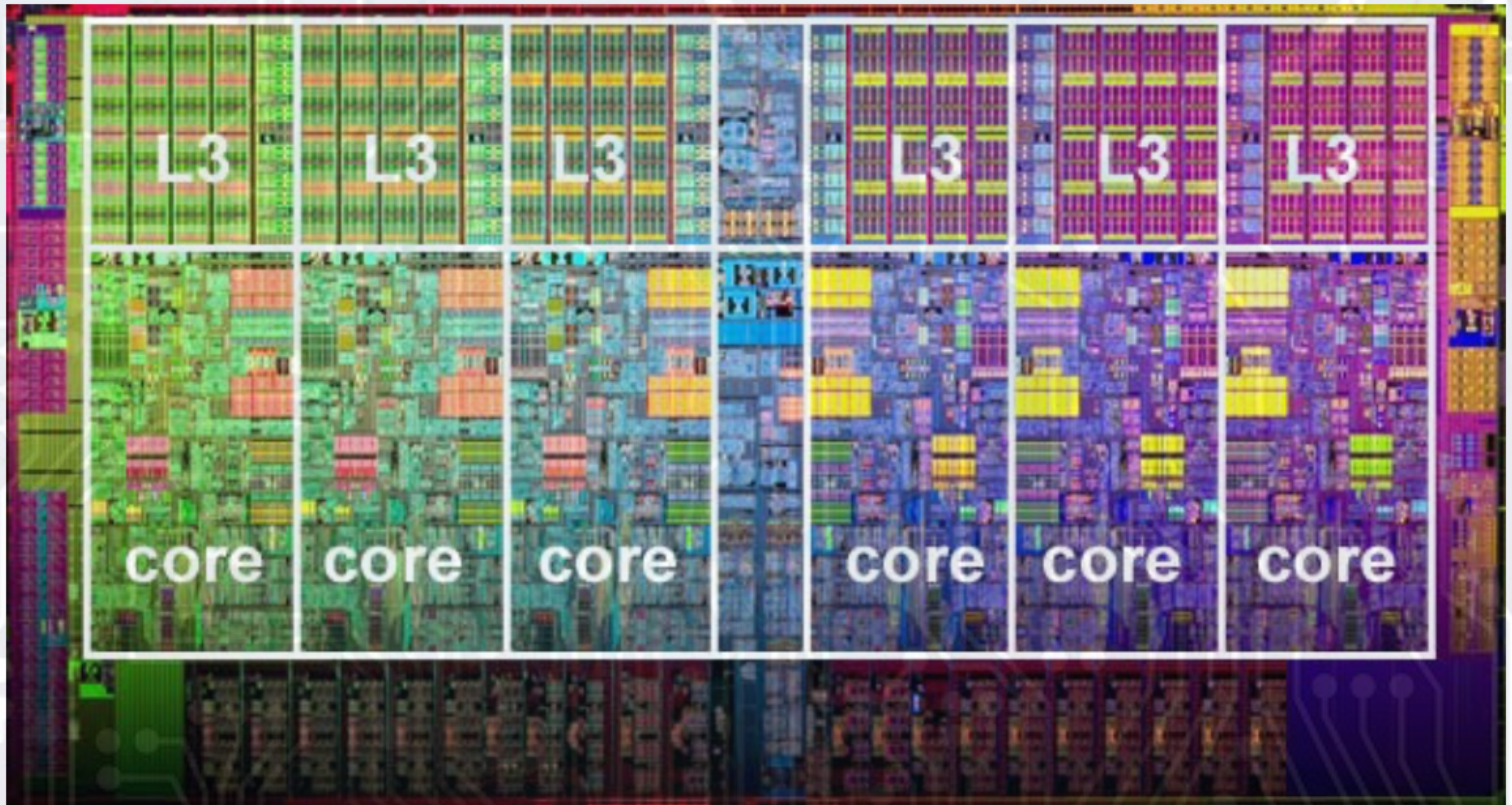


Why use Parallel computing?

Part 1: Lecture 2

Benefits of using Parallel computing

- **MAKE BETTER USE OF UNDERLYING PARALLEL HARDWARE**
  - Modern computers, even laptops, are parallel in architecture with multiple processors/cores.
  - Parallel software is specifically intended for parallel hardware with multiple cores, threads, etc.
  - In most cases, serial programs run on modern computers "waste" potential computing power.

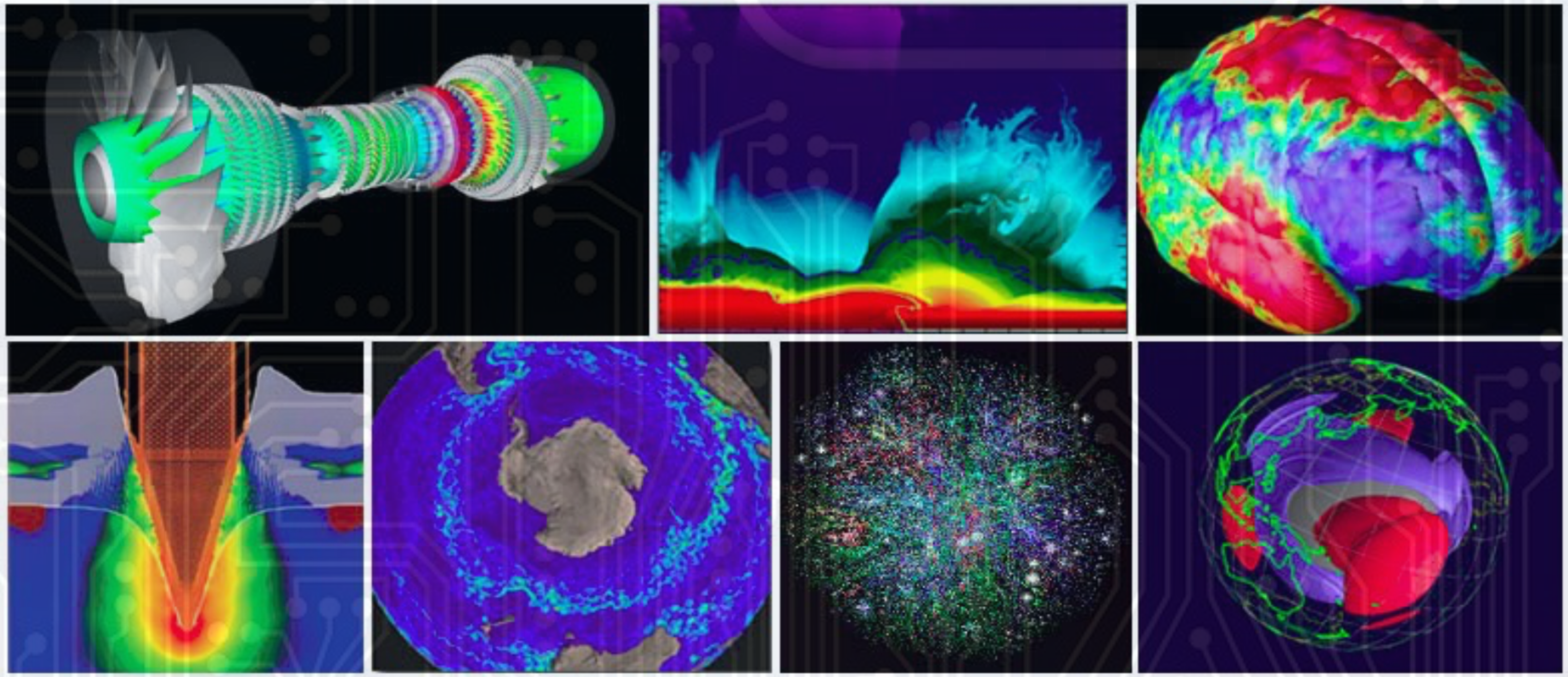


Why use Parallel computing?

Part 1: Lecture 2

Who is using Parallel computing?

- It has been used to model difficult problems in many areas of science and engineering:
  - Atmosphere, Earth, Environment
  - Physics - applied, nuclear, particle, condensed matter, high pressure, fusion, photonics
  - Bioscience, Biotechnology, Genetics

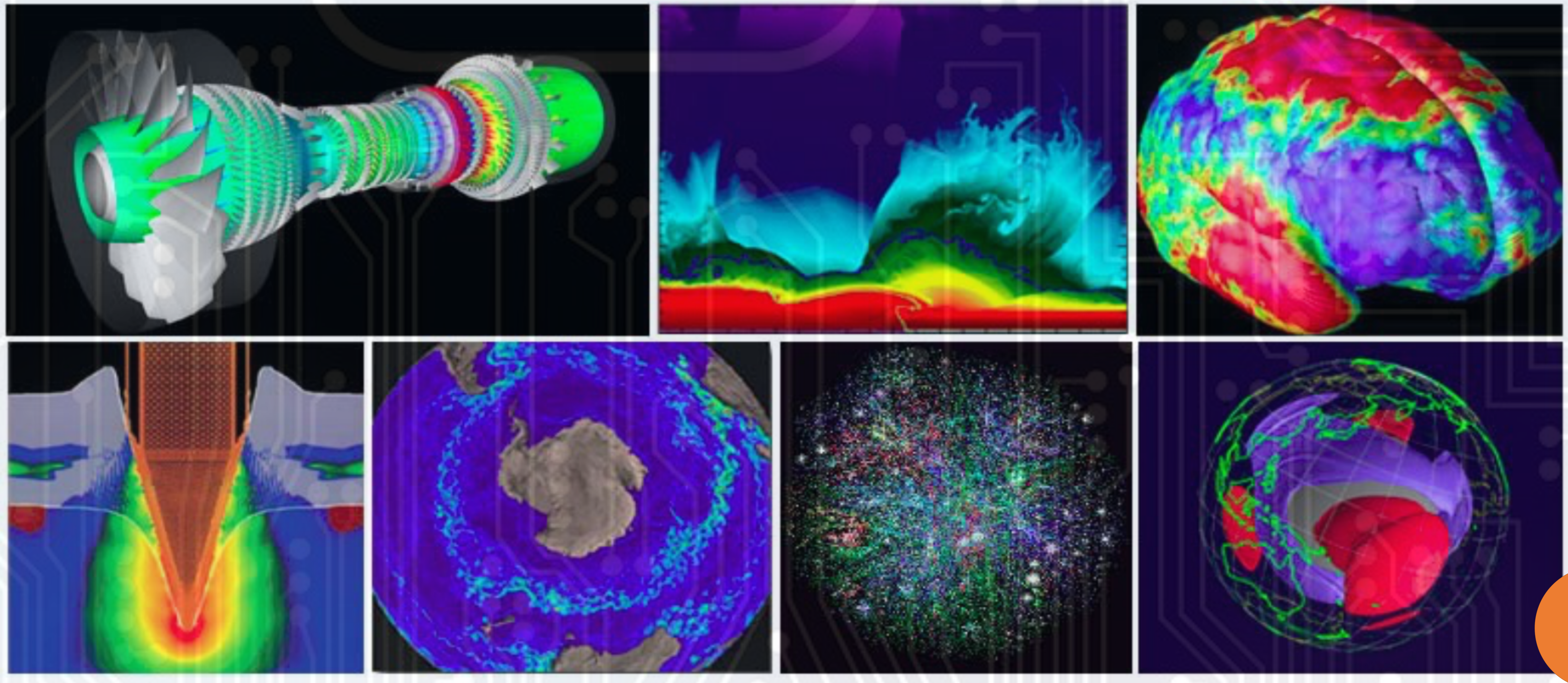


Why use Parallel computing?

Part 1: Lecture 2

Who is using Parallel computing?

- It has been used to model difficult problems in many areas of science and engineering:
  - Chemistry, Molecular Sciences
  - Geology, Seismology
  - Mechanical Engineering - from prosthetics to spacecraft
  - Defense, Weapons etc

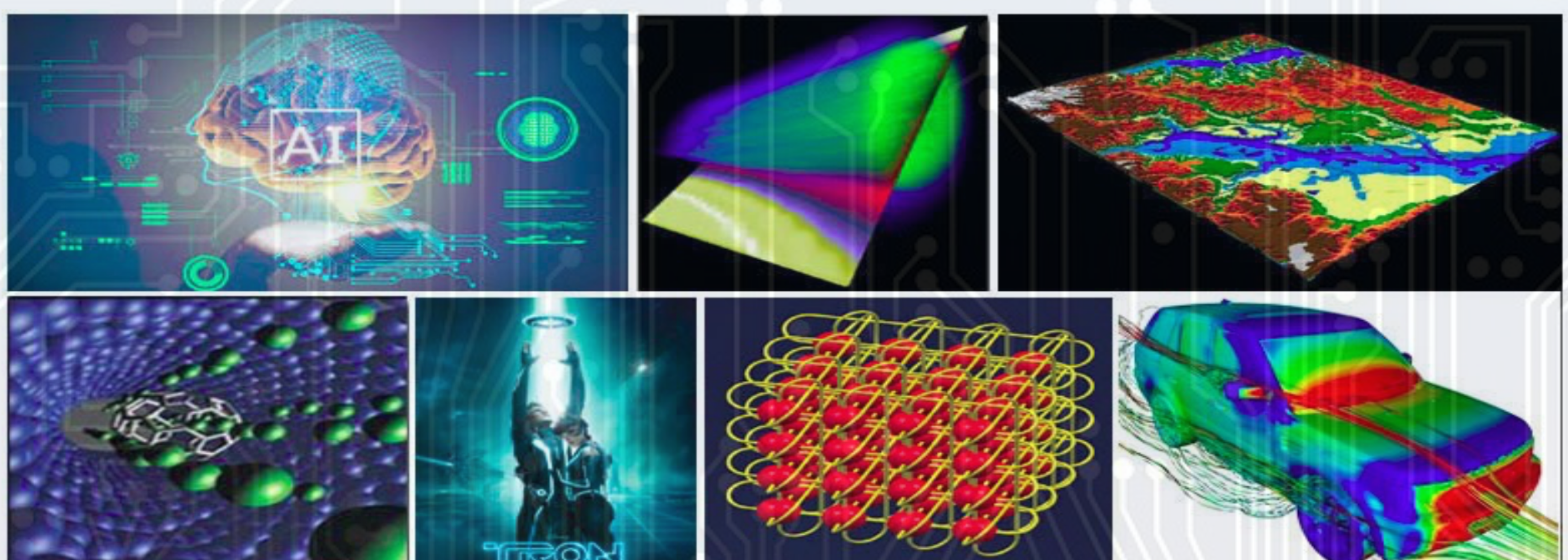


Why use Parallel computing?

Part 1: Lecture 2

Who is using Parallel computing?

- Industrial and commercial applications require the processing of large amounts of data in sophisticated ways. For example:
  - "Big Data", databases, data mining
  - Artificial Intelligence (AI)
  - Oil exploration
  - Medical imaging and diagnosis

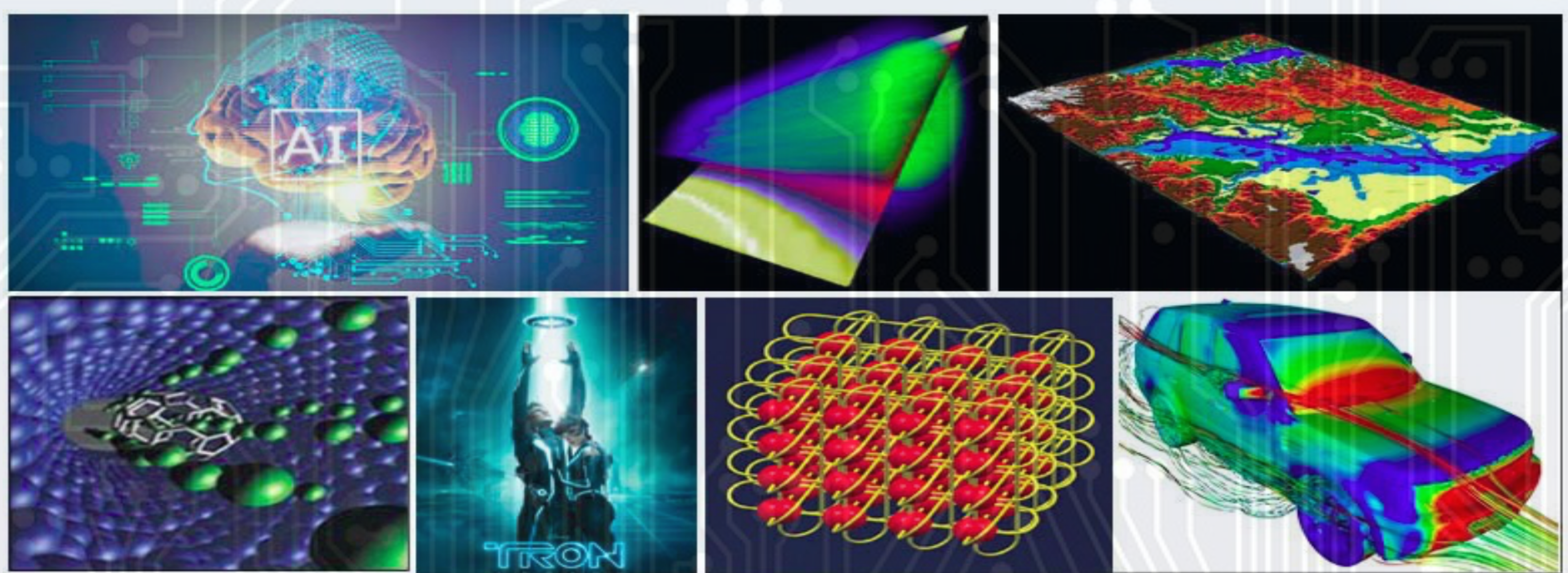


Why use Parallel computing?

Part 1: Lecture 2

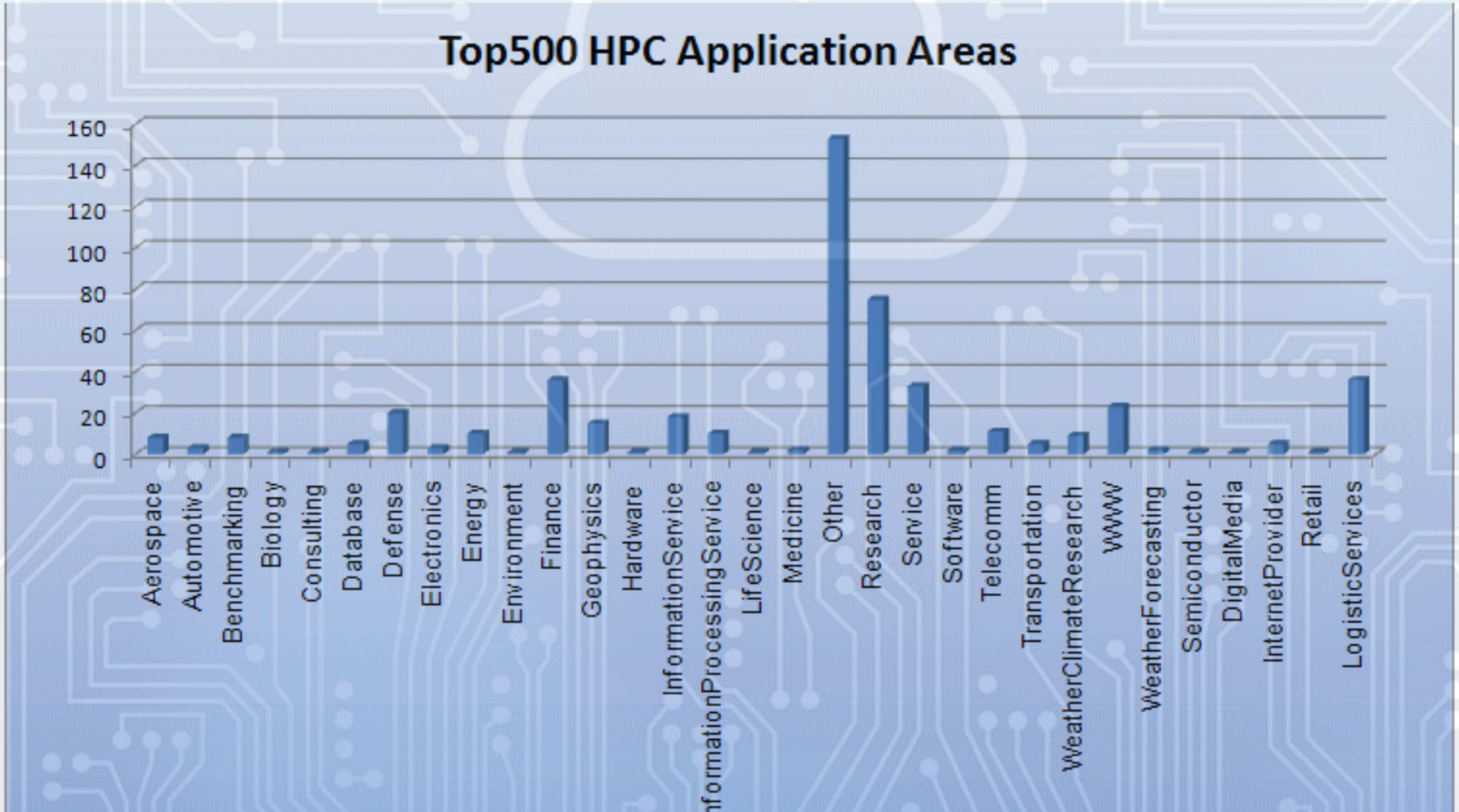
Who is using Parallel computing?

- Pharmaceutical design
- Financial and economic modeling
- Management of national and multi-national corporations
- Advanced graphics and virtual reality, particularly in the entertainment industry
- Networked video and multi-media technologies



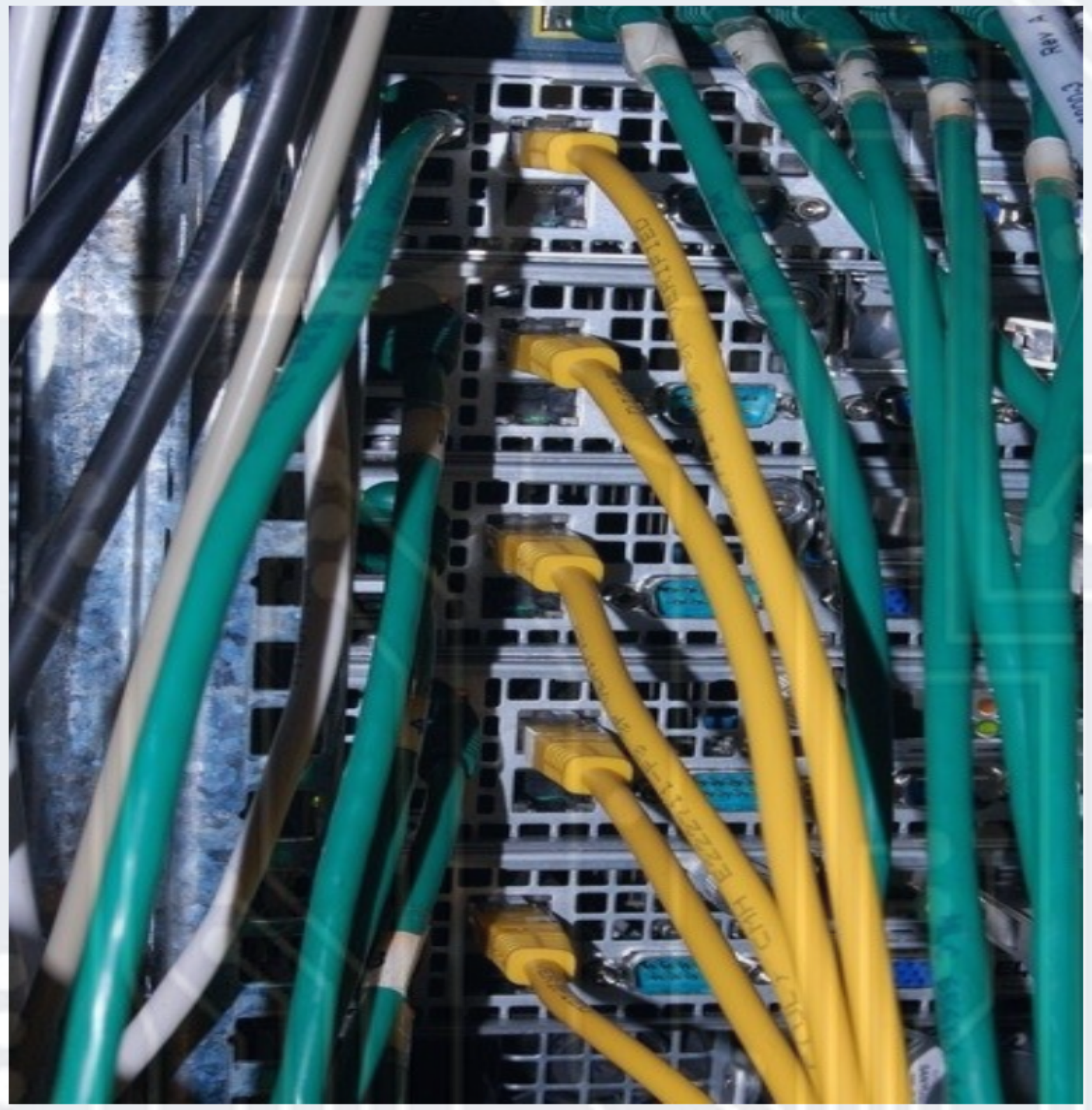
Who is using Parallel computing?

- Global Applications
  - Parallel computing is now being used extensively around the world, in a wide variety of applications.



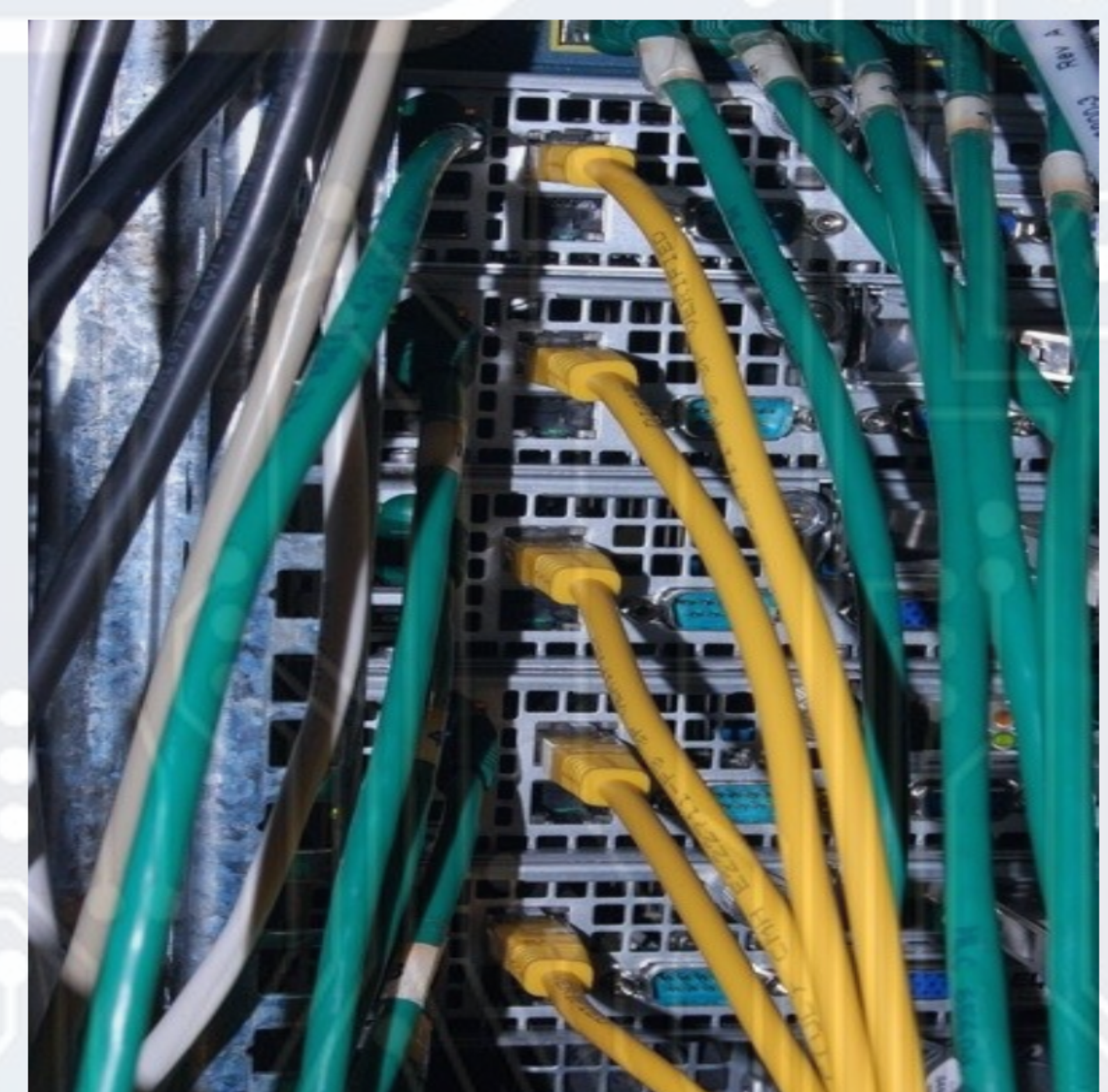
Cluster computing

- Cluster Computing is a computing infrastructure that is comprised of a group of linked computers working together closely in order to improve performance
- Cluster computing will have hybrid or distributed memory architecture and usually designed to mimic MIMD or SIMD parallel program model.



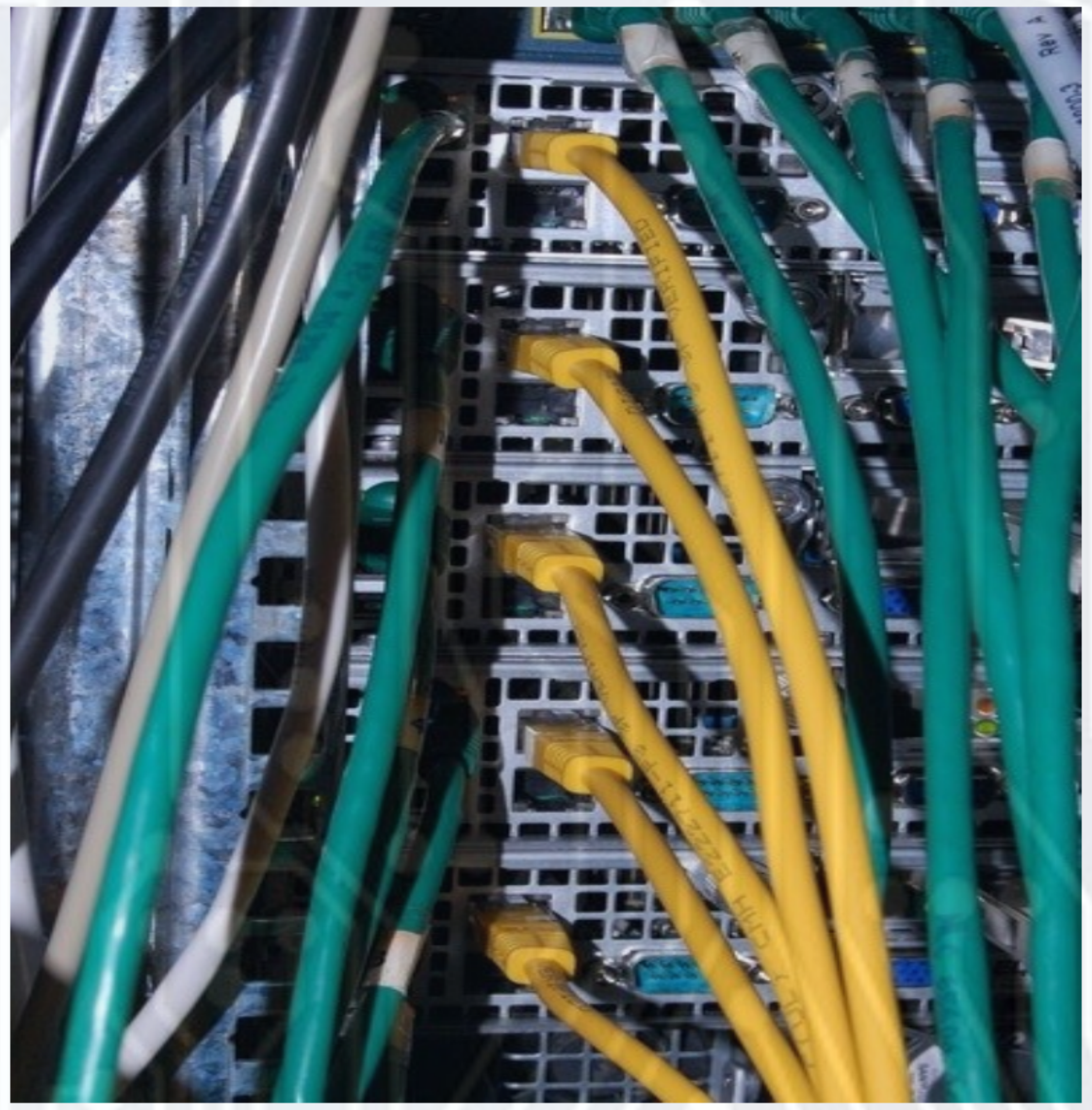
Cluster computing

- The primary reasons for Cluster computing:
  - It improves the performance of the system (wall clock time)
  - It can solve larger problems
  - Provide concurrency (do multiple things at the same time)
  - Enable us to use distributed resources such as lab machines, which save cost, time, and overcome memory constraints



Cluster computing: Factors to measure performance of parallel programs

- **Latency and Bandwidth measure the performance of a cluster environment:**
  1. Bandwidth tells us the rate of data transmission per second from one process to another in a given network
  2. Latency tells how much time will be spent in order to establish a communication link between processes over the network
  3. Latency and bandwidth don't depend on the actual application but on the networking facility



Cluster computing: Factors to measure performance of parallel programs

- It is also possible to measure the performance of a parallel program designed to solve specific application. Such performance depends on several factors such as:
  - Communication requirement of the Application
  - Embarrassingly parallel problem (a problem that couldn’t be affected by the communication facility)
  - Highly communication dependent problem
  - Architecture of cluster(specification of nodes, processors, heterogeneous, etc)
  - The algorithm implemented
  - The compiler used

Parallel computing

Part 1: Lecture 2

Cluster computing: Factors to measure performance of parallel programs

- The performance improvement obtained using an application developed in a cluster environment can be measured using the parameter like
  - Absolute Speedup (S_{an}) = time of equivalent sequential program/ time of parallel program in N-processor(t_{seq}/t_{par})
  - Absolute Efficiency$(E_{an}) = S_{an}/N$
  - Relative Speedup (S_{rn}) = time required for the parallel code in 1 processor/time of parallel program in N-processor
  - Relative Efficiency$(E_{rn}) = S_{rn}/N$

Cluster computing: Factors to measure performance of parallel programs

- **Question:**
  1. What do you say if Absolute Speedup (S_{an}) is less than one?
  2. What do you say if Absolute Efficiency $(E_{an}) = 1$
  3. What do you say if Absolute Efficiency$(E_{an}) > 1$
  4. What do you observe by looking at Relative Speedup (S_{rn}) for different value of N
  5. What do you observe by looking at Relative Efficiency (E_{rn}) for different value of N

Parallel Programming Model

- Parallel programming models exist as an abstraction above hardware and memory architectures
- The models basically different in :
  - How the application divided into parallel tasks(the computational task)
  - How mapping is done between computational task and processing elements
  - How data distribution with in memory handled
  - How mapping of communication between inter-connected network is done
  - How inter-task synchronization is implemented

Parallel Programming Model

- There are several parallel programming models in common use:
  - Shared Memory (without threads)
  - Threads
  - Distributed Memory / message passing
  - Data Parallel
  - Hybrid
  - Single program multiple data (SPMD)
  - Multiple Program Multiple Data (MPMD)

Parallel Programming Model

- Though various parallel programming models exist, in this course we will focus on two of the most common model:
  - OpenMP(Shared Memory Architecture) and
  - MPI(message passing programming models model in a distributed or hybrid memory architecture)

OPEN MPI

OpenMP®

MPI Vs OpenMP: Parallel Programming Model

NO	Factor	OpenMP	MPI
1	Task identification and mapping	Implicit	Explicit
2	Communication mapping	Implcit	Explicit
3	Synchronization	Implcit	Explicit
4	Data Distribution	Implcit (directive based)	Explict(Programer)
5	Environment	SMP	SMP, Distributed or hybrid
6	Implementation	Multi-threading(forking)	Mainly a process on processor

Advantages of MPI

- Does not require shared memory architectures which are more expensive than distributed memory architectures
- Can be used on a wider range of problems since it exploits both task parallelism and data parallelism
- Can run on both shared memory and distributed memory architectures
- Highly portable with specific optimization for the implementation on most hardware

OPEN MPI

Disdvantages of MPI

- Requires more programming changes to go from serial to parallel version
- Can be harder to debug

Advantages of OpenMP

- Considered by some to be easier to program and debug (compared to MPI)
- Data layout and decomposition is handled automatically by directives.
- Allows incremental parallelism: directives can be added incrementally, so the program can be parallelized one portion after another and thus no dramatic change to code is needed
- Unified code for both serial and parallel applications: OpenMP constructs are treated as comments when sequential compilers are used.
- Original (serial) code statements need not, in general, be modified when parallelized with OpenMP. This reduces the chance of introducing bugs and helps maintenance as well.

Disdvantages of OpenMP

- Currently only runs efficiently in shared-memory multi processor platforms
- Require compiler that supports OpenMP
- Scalability is limited by memory architecture
- Reliable error handling is missing.
- Lacks fine-grained mechanisms to control thread-processor mapping
- Synchronization between subsets of threads is not allowed
- Mostly used for loop parallelization
- Can be difficult to debug, due to implicit communication between threads via shared variables

OpenMP®

THANK YOU!