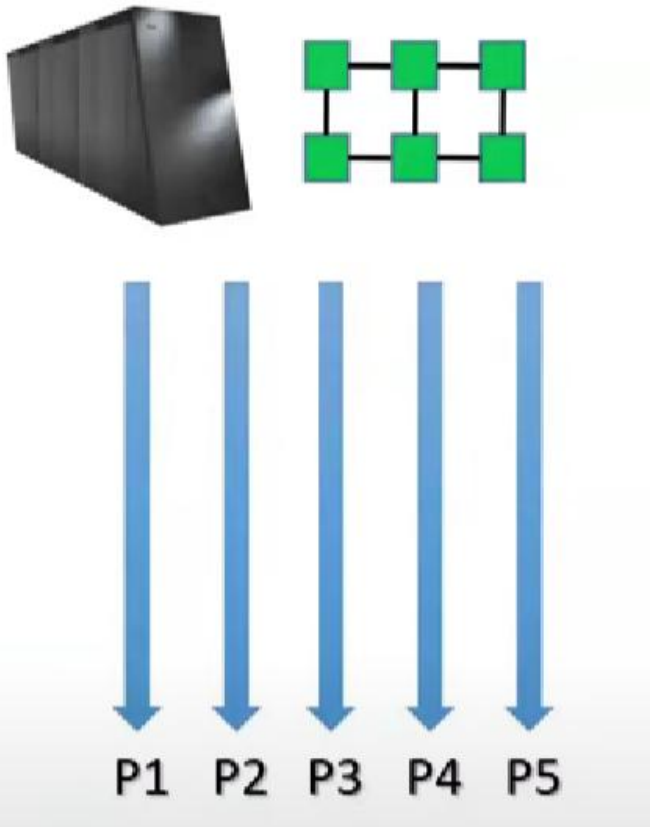
# HIGH PERFORMANCE COMPUTING WITH C

Introduction...

Bijaya Ghimire




# WHAT IS HPC....?

HPC is a technology that uses **clusters of powerful processors** that work in **parallel** to process massive, multidimensional data sets and **solve complex problems** at **extremely high speeds.**

*[1] IBM on 'High performance Computing'*

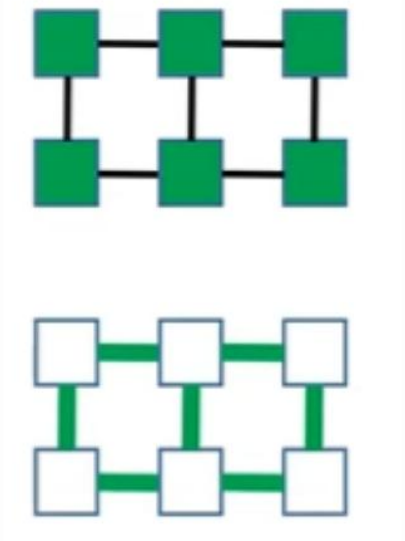
HPC includes work on 'four basic building blocks'(*general*):

- Theory (numerical laws & Algorithm, Performance Models, Complexity & Efficiency etc.)
- Technology (Multi-core & Many-core Processors, Supercomputers & Clusters, networks, storages, etc.)
- Architecture (shared-memory, distributed-memory, Hybrid Architectures, interconnects, etc.)
- Software (libraries & Frameworks (CUDA), Schedulers & Resource Managers, Monitoring & Debugging Tools, Applications, etc.)

*[2] Introduction to High Performance Computing for Scientists and Engineers*

# PARALLEL COMPUTING | HPC | SUPER COMPUTER

**Parallel Computing (Technique):** Breaking a large task into smaller parts and run simultaneously on multiple processor/core
**HPC (System + Practice):** HPC is the field that applies parallel computing to solve real-world, large problems.
**Supercomputer (Infrastructure):** Specialized hardware built to run HPC workloads.



**Does HPC Apply Only in Supercomputers?**
Short Answer: ❌ No.

- HPC is a practice, not tied only to supercomputers.
- It means using parallel computing + high-performance hardware/software to solve large problems fast.
- Supercomputers are just the highest-end machines for HPC.
- **Can be practice on Single-node HPC with many-core GPUs(Laptops, PCs), Services like AWS Parallel Cluster, Microsoft Azure HPC, Google Cloud HPC, University/research labs often connect multiple commodity PCs → "Beowulf clusters."**

# FIELD OF APPLICATIONS

### 🧬 Healthcare, Genomics & Life Sciences

HPC speeds up genome sequencing, drug discovery, cancer diagnosis, and molecular modeling, turning tasks that once took years into hours.

### 🛰️ Government & Defense

Used for weather forecasting, climate modeling, energy research, and intelligence analysis, HPC supports disaster preparedness and security.

### 🎥 Media & Entertainment

It enables fast 3D rendering, special effects, and smooth live streaming, cutting production costs and time.

### ⚡ Energy

HPC processes seismic data, simulates reservoirs, and models wind and terrain to optimize energy exploration and renewable design

### 💰 Banking & Financial Services

HPC powers automated trading, fraud detection, and risk analysis through large-scale simulations for real-time decisions.

### 🚙 Automotive Industry

It helps design safer, more efficient cars with crash tests, aerodynamics, and battery simulations, reducing prototype costs.

### 🔒 Cybersecurity

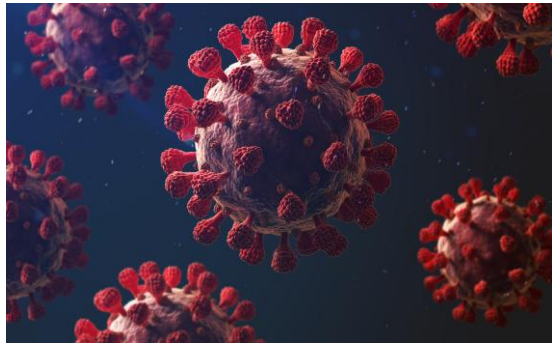
HPC analyzes massive network data in real time, detecting threats and preventing cyberattacks before they spread.

### 🧠 Deeplearning

Enables training of massive models with billions of parameters on huge datasets. By using thousands of CPUs/GPUs and high-speed networks, HPC reduces training time from months to days and powers breakthroughs like GPT and AlphaFold.

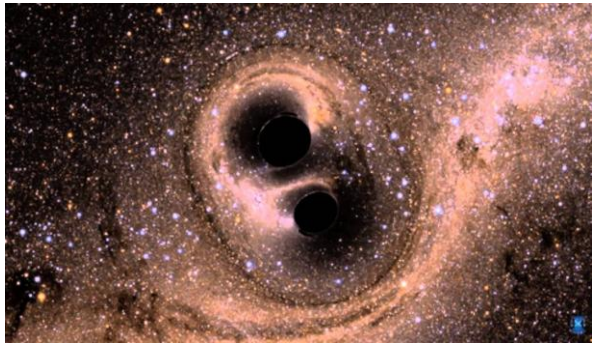
*[3]IBM. (n.d.). What is high performance computing? IBM. https://www.ibm.com/think/topics/hpc*

# SOME PROBLEM SOLVED USING HPC



COVID-19 Vaccine Development

Weather and Climate Change Prediction



Black Hole & Gravitational Waves

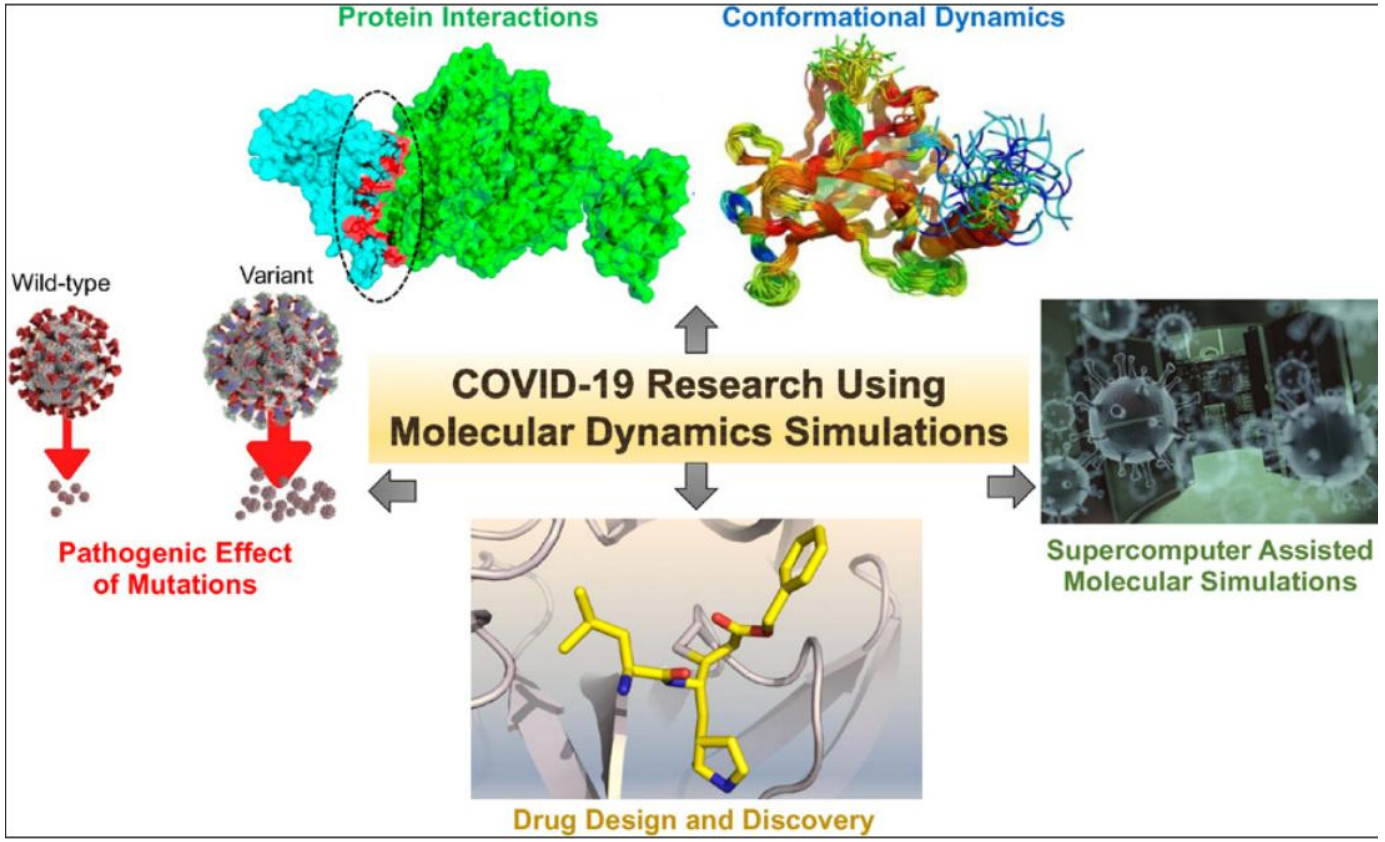
And Many More.....

## 💉 COVID-19 Vaccine Development with HPC

**Challenge:**
Traditional drug discovery takes years too slow for a global pandemic.

**HPC Role:**

- Summit supercomputer screened **8,000+ compounds in days**.
- Ran **molecular simulations** to study virus–cell interactions.
- Analyzed **genomic data** to track mutations.




**Impact:**

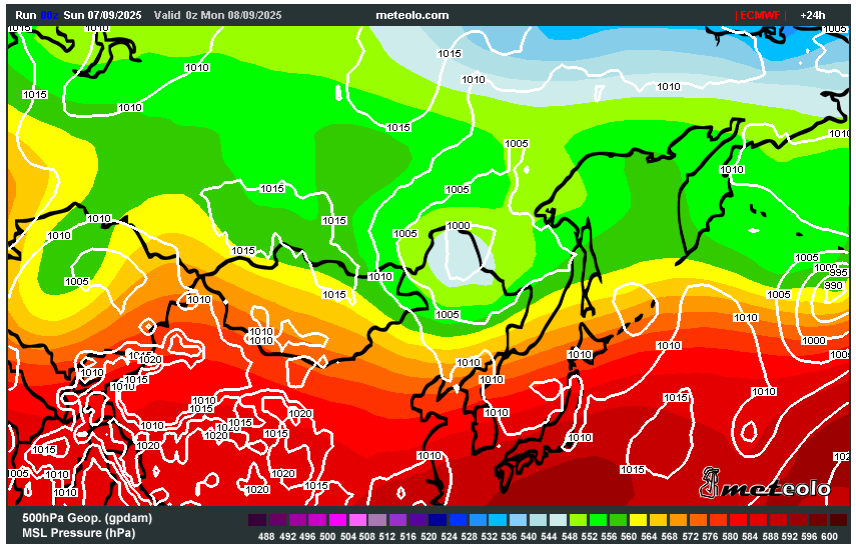
- Reduced discovery timeline from **years → months**.
- Helped develop Pfizer, Moderna, and AstraZeneca vaccines.
- Enabled vaccines by **end of 2020**, saving millions of lives.


Aditya K. Padhi, Soumya Lipsa Rath, and Timir Tripathi, **Accelerating COVID-19 Research Using Molecular Dynamics Simulation,** *The Journal of Physical Chemistry B* **2021** *125* (32), 9078-9091
DOI: 10.1021/acs.jpcb.1c04556

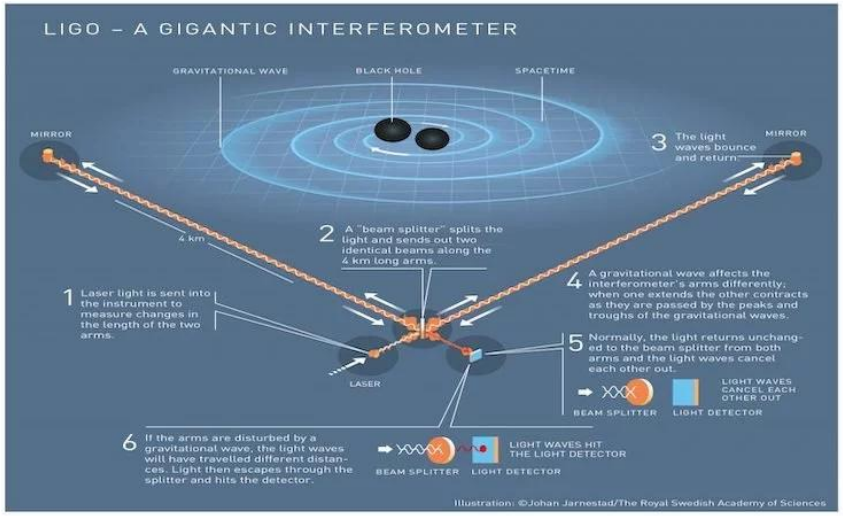
# 🌎 HPC in Climate & Space

## 🌦️ Climate Modeling & Weather Forecasting

- Simulates Earth's atmosphere, oceans, and land with billions of equations.
- Used by ECMWF & Fugaku supercomputers.
- Predicts **hurricanes, floods, and climate change** ahead.
- Saves lives and informs global policy.




Meteolo. (n.d.). *ECMWF Asia*. Meteolo. https://meteolo.com/asia/ecmwf/ecmwf-asia#google_vignette




OSR. (2022, September 26). *Observing gravitational waves at LIGO*. OSR. https://osr.org/blog/kids/observing-gravitational-waves-at-ligo/

## 🌌 Black Holes & Gravitational Waves

- HPC simulate Einstein's equations for massive cosmic events.
- Simulated **black hole collisions**.
- Supported **LIGO's 2015 discovery** of gravitational waves.
- Helps us understand the universe's most extreme phenomena.

# Working Mechanism of HPC

**Problem Breakdown**
Divide a large scientific problem (e.g., weather simulation) into smaller tasks.

**Parallel Processing**
Thousands of CPUs/GPUs work simultaneously on different tasks.

**High-Speed Interconnection**
Processors communicate using ultra-fast networks (InfiniBand, NVLink).

**Memory & Storage Management**
Massive memory and parallel file systems (Lustre, GPFS) handle huge datasets

**Scheduling & Resource Allocation**
Schedulers (SLURM. PBS, LSF) assign resources efficiently

**Execution & Aggregation**
Processors compute tasks, results are combined into a full solution

- Cloud VMs/GPUs
- Thousands of **nodes with CPUs/GPUs**
- **OpenMP (CPU multicore)** or **CUDA/OpenCL (GPU)** parallelizes tasks.[Single Node]

- Virtual high-speed networks (e.g., Azure InfiniBand, AWS EFA)
- **Dedicated low-latency interconnects** (InfiniBand, NVLink, Omnipath).
- No external networking

- Data lives in **cloud object storage** (S3, Blob) with parallel file access.
- **Distributed memory per node**, parallel file systems
- Shared memory access

- Cloud job schedulers auto-scale resources on demand.
- Cluster schedulers (SLURM, PBS, LSF) allocate nodes and resources.
- OS or OpenMP runtime handles thread scheduling. CUDA runtime schedules GPU kernels.

# So, Basically, We need granular access across our Nodes/Cores, Storage Devices(Memory), Resources Managers(Schedulers)

This is why we use **C** programming Language

**1. Performance & Speed** 🚀 –
C compiles close to machine code with minimal overhead, making it ideal for fast HPC tasks.

**2. Control Over Memory & Hardware** ⚙️
C provides fine control of memory and hardware, allowing deep performance optimization.

**3. Portability** 🌎
C runs on all platforms, from laptops to the world's largest supercomputers.

**4. Foundation for HPC Libraries** 📚
Most HPC libraries (MPI, OpenMP, CUDA, OpenCL) are built with C or expose C interfaces.

**5. Interoperability** 🔗
C easily integrates with Fortran, C++, Python, and GPU code, often acting as the glue in HPC workflows.

**6. Proven in HPC History** 🏛️
Many legacy and modern HPC codes are written in C, proving its reliability in supercomputing.

# 🖥️ DIFFERENT PROGRAMMING LANGUAGES IN HPC

1: Fortran

2: C/C++

3: Julia

4: Python

5: Matlab

6: R

⚖️ **Rule of Thumb**

- **For speed-critical kernels:** C, C++, Fortran, CUDA/OpenCL.
- **For orchestration & data analysis:** Python, R, Julia, MATLAB.
- **For legacy scientific domains:** Fortran (still strong in weather, CFD, physics).
- **For modern ML/AI HPC:** Python (front-end) + C/CUDA (back-end).

Rust is emerging as a **modern alternative to C/C++** for HPC kernels where **safety + speed** both matter.

# Setting Up: C compiler and VsCode

- Install VS Code → Download from code.visualstudio.com.
- Install C/C++ Extension → Open Extensions (Ctrl+Shift+X ) → search C/C++ (Microsoft) → Install.
- Optional Extension: Install Code Runner for quick execution.

**Windows Setup** 🪟

Install Compiler:

Option A: MinGW-w64 (recommended) → Download from winlibs.com.

Option B: MSYS2 → More advanced package manager.

Add Compiler to PATH: Find the bin folder of MinGW (e.g., C:\mingw64\bin)

Add it to System Environment Variables → Path.

Verify Installation: *gcc –version (*If it shows version info, you're set.)

**macOS Setup** 🍏

Install Xcode Command Line Tools:

*xcode-select --install* (This installs clang, the default C compiler on macOS.)

Verify Compiler: *clang --version | (gcc --version)*

Use VS Code → Create program.c → Compile in terminal:

*clang program.c -o program | (gcc program.c -o program)*

*./program*

**Linux Setup** 🐧

Install GCC Compiler:

Ubuntu/Debian

*sudo apt update*

*sudo apt install build-essential*

Verify Compiler:

*gcc –version*

Use VS Code → Create program.c → Compile in terminal:

*gcc program.c -o program*

*./program*

# Tasks to be done Before Workshop

1. Installing C and setup VsCode
2. Revise the lecture slides and be familiar with following topics:
    - Preprocessor directives
    - The main() function
    - Input / Output
    - The printf() function
    - The scanf() function
    - Arrays
    - Strings in C
    - Function prototypes
    - Structures

*Please refer to the lecture slides…..*

Cheat sheet for C and the recommended Book "*The C Programming Language (2nd Edition) Brian Kernighan and Dennis Ritchie Prentice Hall, 1988 ISBN 978-0131103627* " is available in MST under my resource.

# ANY QUESTIONS……?