



**Welcome to this session:**  
**Skills Bootcamp - High Performance**  
**Computing for Data Science (Practical**  
**Applications and Implementation)**

**The session will start shortly...**

Questions? Drop them in the chat.  
We'll have dedicated moderators  
answering questions.



# Skills Bootcamp Data Science Housekeeping

---

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. We will be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

# Skills Bootcamp Data Science Housekeeping

---

- For all **non-academic questions**, please submit a query: [www.hyperiondev.com/support](http://www.hyperiondev.com/support)
- Report a safeguarding incident: [www.hyperiondev.com/safeguardreporting](http://www.hyperiondev.com/safeguardreporting)
- We would love your feedback on lectures: [Feedback on Lectures.](#)
- Find all the lecture **content** in your [Lecture Backpack](#) on GitHub.
- If you are hearing impaired, kindly use your computer's function through Google chrome to enable captions.

# Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:



Ian Wyles  
Designated Safeguarding  
Lead



Simone Botes



Nurhaan Snyman



Rafiq Manan



Ronald Munodawafa



Tevin Pitts

Scan to report a  
safeguarding concern



or email the Designated  
Safeguarding Lead:  
Ian Wyles

[safeguarding@hyperiondev.com](mailto:safeguarding@hyperiondev.com)

# Skills Bootcamp Progression Overview

## ✓ Criterion 1 - Initial Requirements

Specific achievements **within the first two weeks** of the program.

To meet this criterion, students need to, by no later than **01 December 2024 (C11)** or **22 December 2024 (C12)**:

- **Guided Learning Hours (GLH):** Attend a **minimum of 7-8 GLH per week** (lectures, workshops, or mentor calls) for a total minimum of **15 GLH**.
- **Task Completion:** Successfully complete the **first 4 of the assigned tasks**.

## ✓ Criterion 2 - Mid-Course Progress

Progress through the successful completion of tasks **within the first half** of the program.

To meet this criterion, students should, by no later than **12 January 2025 (C11)** or **02 February 2025 (C12)**:

- **Guided Learning Hours (GLH):** Complete at least **60 GLH**.
- **Task Completion :** Successfully complete the **first 13 of the assigned tasks**.

# Skills Bootcamp Progression Overview

## ✓ Criterion 3 – End-Course Progress

Showcasing students' progress nearing the completion of the course.

To meet this criterion, students should:

- **Guided Learning Hours (GLH):** Complete the **total minimum required GLH**, by the **support end date**.
- **Task Completion : Complete all mandatory tasks**, including any necessary resubmissions, by the end of the bootcamp, **09 March 2025 (C11)** or **30 March 2025 (C12)**.

## ✓ Criterion 4 - Employability

Demonstrating progress to find employment.

To meet this criterion, students should:

- **Record an Interview Invite:** Students are required to record proof of invitation to an interview by **30 March 2025 (C11)** or **04 May 2025 (C12)**.
  - **South Holland Students** are required to proof and interview by **17 March 2025**.
- **Record a Final Job Outcome :** Within 12 weeks post-graduation, students are required to record a job outcome.



## Learning Outcomes

- ❖ **Discuss the role of High-Performance Computing (HPC)** in data science and its advantages over traditional computing.
- ❖ **Explain key concepts in HPC**, including parallel and distributed computing, GPU acceleration, and cloud-based HPC solutions.
- ❖ **Evaluate different HPC architectures** and frameworks used in data science, such as MPI, CUDA, and Spark.
- ❖ **Analyse real-world case studies** where HPC has been used to solve complex data science problems.
- ❖ **Recognize challenges and limitations of HPC**, including scalability, energy efficiency, and cost.

# Lecture Overview

---

- Introduction to HPC
- Key Concepts
- Frameworks
- Case Studies in HPC
- Challenges and Future Trends in HPC







# What are the benefits of using High-Performance Computing in data science?

# Large-Scale Data Processing

As data science applications become more complex, traditional computing methods struggle to handle large-scale data processing efficiently. Imagine that we could share the workload involved in a project between multiple devices.

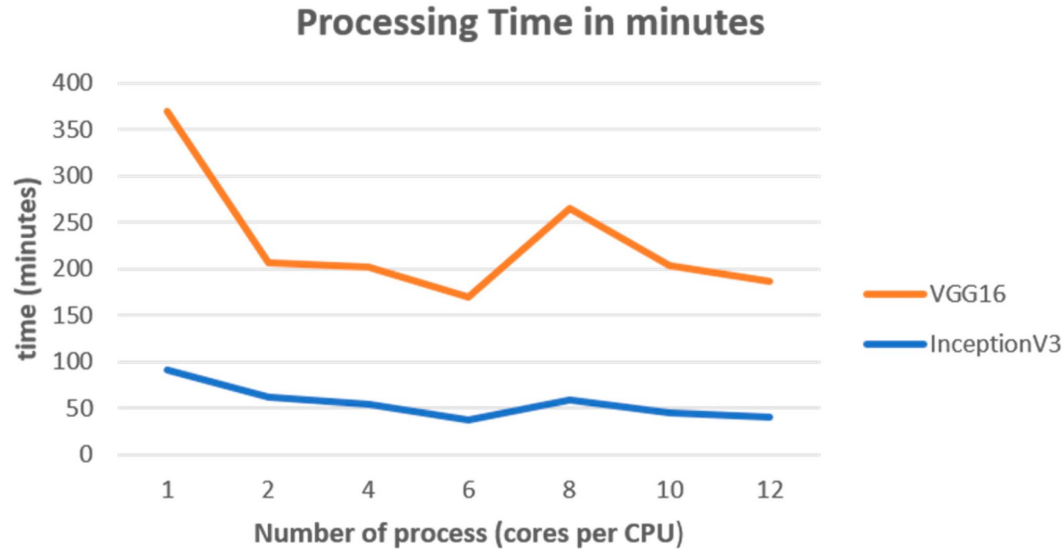
- What do we need to consider regarding the **division of work**?
- How will transitioning a problem from being solved by one device, to multiple devices simultaneously **affect the complexity of the problem** and the **cost to the environment**?

# Large-Scale Data Processing

High-Performance Computing (HPC) provides powerful solutions, but it comes with its own set of challenges and considerations. HPC plays a crucial role in areas such as climate modeling, genomic analysis, financial forecasting, and AI research. Understanding how to leverage HPC is essential for data scientists tackling large-scale problems.

- *How long do you think it would take to train an AI model on a single CPU versus a distributed HPC system?*

# Large-Scale Data Processing



VGG16 and InceptionV3 are two different Convolutional Neural Network models



Source: [Weed Classification using Convolutional Neural Networks](#)

# High Performance Computing

The use of powerful processors, networks and parallel supercomputers to tackle problems that are very computationally or data-intensive

- ❖ A HPC system **divides workloads into smaller tasks** and assigns them to **multiple resources** for **simultaneous processing**.
- ❖ HPC systems can solve problems that are either **too large** for standard computers to handle individually or would take **too long to process**.
  - For this reason, it is also sometimes referred to as **supercomputing**.



- 
- 
- ❖ HPC's parallel computing capabilities can greatly **accelerate iterative processes** compared to traditional computing.
    - HPC systems typically run at speeds **more than one million times faster** than the fastest commodity desktop, laptop or server.
  - ❖ HPC's ability to **quickly process massive amounts of data** powers some of the most fundamental aspects of today's society.
    - Reduce the time to **train deep learning models** from days to hours
    - Banks can verify **fraud** on millions of credit card transactions at once
    - Automakers test their car designs for **crash safety**
    - Simulations to help us **predict the weather or molecular structure**

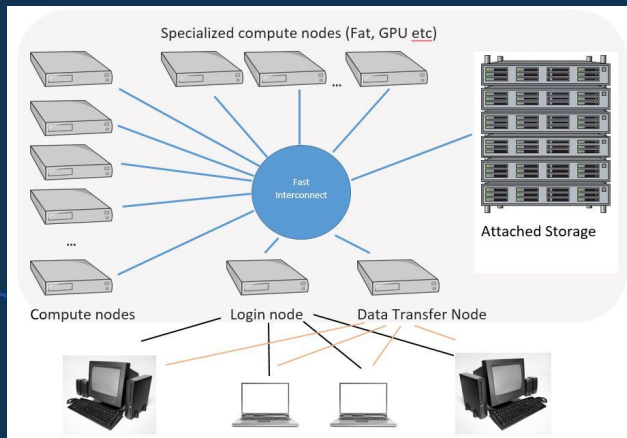


# Key Concepts in HPC

- ❖ HPC can take the form of **custom-built supercomputers** or **groups of individual computers** called **clusters or nodes**.
- ❖ **An HPC cluster** comprises multiple **high-speed computer servers** which use **high-performance multi-core CPUs** or **GPUs**.
- ❖ GPUs are **specialized computer chips** designed to process **large amounts of data** in parallel, making them ideal for some HPC, and are currently the standard for ML/AI computations.
- ❖ A single HPC cluster can include 100,000 or more nodes.

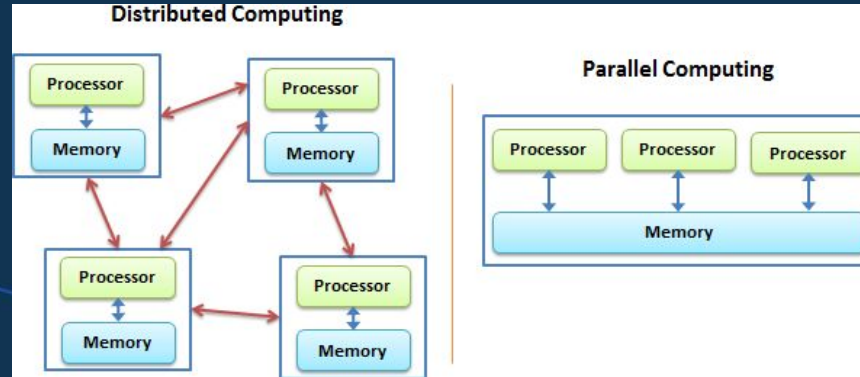
# Key Concepts in HPC

- ❖ High performance computing has **three main components**:
  - Compute, Network, Storage
- ❖ The nodes (**compute**) of the HPC system are connected to other nodes to run algorithms and software simultaneously, and are then connected (**network**) with data servers (**storage**) for output.



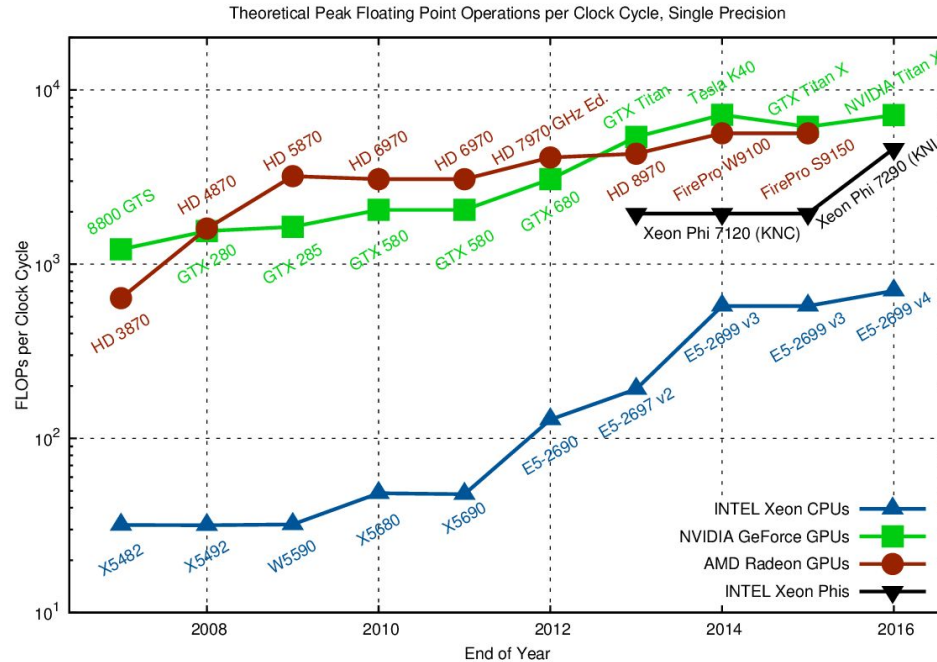
Source: [Iowa State University](#)

- ❖ HPC can typically be broken down into two general design types:
  - **Cluster or parallel computing:** runs multiple tasks simultaneously on multiple, closely-connected processors. Best for computationally complex problems.
  - **Distributed computing:** connects computers via a network so that they can act as one powerful machine. Suits problems with massive datasets.



Source: [O'Reilly](#)

# CPU vs GPU for ML



Source: [Comparing CPUs and GPUs for Machine Learning](#)

# Let's Breathe!

Let's take a small break  
before moving on to  
the next HPC.



# HPC Frameworks

- ❖ There are three main programming paradigms/frameworks for HPC:

OpenMP	CUDA	MPI
Open Multi-Processing	Compute Unified Device Architecture	Message-Passing Interface
Shared-memory parallel programming model	Parallel computing platform for NVIDIA GPUs	Message-passing interface for distributed computing
Easy to use and integrate into existing code	Based on threads for parallel execution	Suitable for communication between multiple processing units
Suitable for multi-core processors	Suitable for massive parallel processing	Suitable for distributed computing environments

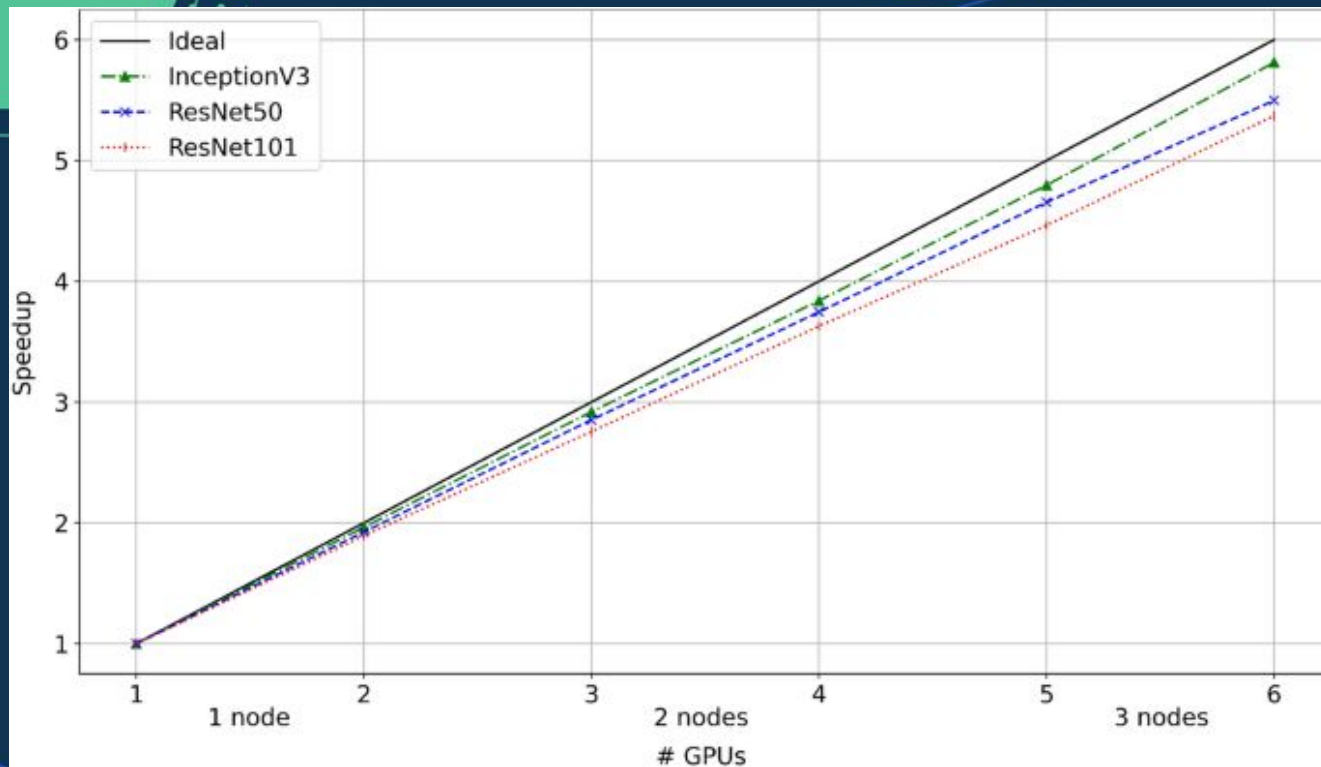


# HPC Frameworks

- ❖ We will often find ourselves in scenarios where we'd like to implement parallel or distributed computing in our code.
- ❖ There are many packages which we can use to implement it in our code:
  - **Spark:** open-source, distributed processing system used for big data workloads.
  - **Hadoop:** framework that allows for the distributed processing of large data sets across clusters of computers.
  - **Dask:** Python module and Big-Data tool that enables scaling pandas and NumPy.
  - More on this in the next lecture...

# Case Studies

- ❖ **AI & Machine Learning:** The paper “[A Container-Based Workflow for Distributed Training of Deep Learning Algorithms in HPC Clusters](#)” presents a methodology that leverages containerization and the Horovod library to facilitate distributed training of deep learning models across multiple GPUs in HPC environments. This approach addresses challenges related to multi-user HPC systems and varying cluster configurations, offering a scalable and efficient solution for training complex models.



Speedup plot for the TensorFlow Benchmark experiment using udocker over the different GPU configurations

# Case Studies

- ❖ **Climate Science:** In “[ExtremeWeather: A Large-Scale Climate Dataset for Semi-Supervised Detection, Localization, and Understanding of Extreme Weather Events](#)”, researchers introduce a comprehensive climate dataset designed to enhance the detection and analysis of extreme weather phenomena. The study demonstrates that semi-supervised learning techniques applied to this dataset can improve the identification and understanding of extreme weather patterns, aiding in more accurate predictions.

# Case Studies

- ❖ **Genomics:** The dissertation “[High-Performance Computing in Next-Generation Sequencing Read Alignment](#)” explores innovative computational strategies and hardware acceleration to optimize the alignment process of short DNA sequences to reference genomes. By integrating HPC techniques, the study addresses the challenges posed by the massive data output of next-generation sequencing, enhancing the efficiency and accuracy of genomic analyses.

# Case Studies

- ❖ **Finance:** The study “[Evaluating the Efficacy of Machine Learning Models in Credit Card Fraud Detection](#)” systematically evaluates the performance of various machine learning algorithms in detecting fraudulent credit card transactions. Utilizing a dataset of over 555,000 transactions, the research compares models such as Logistic Regression, Support Vector Machines, Random Forest, Gradient Boosting, and Neural Networks. The findings highlight that ensemble learning methods and neural networks, when powered by HPC, significantly enhance the accuracy and reliability of fraud detection systems.



# Challenges

- ❖ Although HPCs are undoubtedly useful, they do face challenges:
  - **Cost:** HPC system requires a large budget for hardware, data center, and skilled technicians.
  - **Power consumption:** Newer processors require more energy and generate more heat. High-density CPUs and GPUs used in AI applications consume a lot of power.
  - **Infrastructure complexity:** Legacy data centers may not be able to support the demands of HPC. Integrating multiple processors and accelerators increases system complexity



# Future Trends

- ❖ In the future, HPC systems may leverage quantum computing to achieve unprecedented processing power.
- ❖ Quantum algorithms create multidimensional computational spaces that are a much more efficient way of solving complex problems.
- ❖ As the capacity of HPC processing continues to expand, so too will the ability of systems to tackle our most complex engineering, scientific, and AI-related challenges.



# Why is Apache Spark preferred for big data processing in HPC?

- A. It supports deep learning models
- B. It efficiently handles large-scale distributed data processing
- C. It replaces the need for databases
- D. It is the only available HPC framework



# Why is Apache Spark preferred for big data processing in HPC?

- A. It supports deep learning models
- B. It efficiently handles large-scale distributed data processing**
- C. It replaces the need for databases
- D. It is the only available HPC framework

## Summary

---

- ★ High-Performance Computing (HPC) is essential for large-scale data science applications.
- ★ Parallel and distributed computing techniques enable efficient data processing.
- ★ GPUs and frameworks like Apache Spark accelerate big data analysis.
- ★ HPC is used in AI, genomics, climate science, and finance.
- ★ Future trends include quantum computing and energy-efficient architectures.

# CoGrammar

## Q & A SECTION

**Please use this time to ask  
any questions relating to the  
topic, should you have any.**



# Thank you for attending



**CoGrammar**



Department  
for Education