# 課程回顧（一）

國立臺灣師範大學物理學系 陳俊明

chunming@ntnu.edu.tw

# 什麼是 HPC？

**High Performance Computing(HPC)**

泛指可在較短的時間內計算一般電腦需長時間運算的複雜系統，以提高應用能力，以解決科學、工程或商業中的大問題。

- 計算能力(GFlops)
- 速度(時間)
- 能源的消耗(kW)
  - Energy Efficiency: GFlops/W
  - PUE(Power Usage Effectiveness) = (Total Usage) / (IT Usage)

# 平行計算的分類

- **多核心處理計算 (Multi-core computing)**：一個處理器內有多個處理單元 (processing units) 或稱核心 (core)
- **同時多執行緒 (Simultaneous multithreading)** 又稱超執行緒 (hyper-threading)。
- **對稱多處理 (Symmetric multiprocessing)**：電腦有多個處理器共用主記憶體。例如：2 way, 4 way, 8 way
- **分散式運算 (Distributed computing)**：位於不同地點的電腦透過網路相互連接傳遞訊息與通訊後，並協調它們的行為以達成共同目標而形成的系統。
- **大規模並行運算(Massively parallel computing)**：是由多個微處理器，局部存儲器及網路構成的節點 (node) 組成的並行計算系統；節點間以高速網路。大規模並行處理是一種異步的多指令及數據，因為它的程序有多個程序 (process)，它們分布在各個微處理器上，每個程序有自己獨立的地址空間，程序之間以訊息傳遞介面 (MPI) 進行相互溝通。
- **叢集運算 (Cluster computing)**：一組連接在一起工作的電腦。由於這些電腦協同工作，在許多方面它們可以被視為單個系統。與網格電腦不同，電腦叢集將每個節點設定為執行相同的任務，由軟體控制和排程。

# TOP 500

- 美國橡樹嶺國家實驗室 (Oak Ridge) 的 Frontier 在 TOP 500 目前排第一名。

  (https://www.top500.org/system/180047/) - AMD EPYC 64C
- 日本的理化學研究所 (RIKEN) 與富士通 (Fujitsu) 共同合作打造的 Fugaku (富岳) 在 TOP 500 目前排行第二，曾在 2020 年 6 月 至 2022 年 5 月期間，排行第一 。
  (https://www.top500.org/system/179807/) - ARM A64FX
- Top 500
  - 世界前 500 名 HPC 效能排名
  - https://www.top500.org/
- Green 500
  - 世界前500名HPC的節能排名
  - https://www.top500.org/green500/



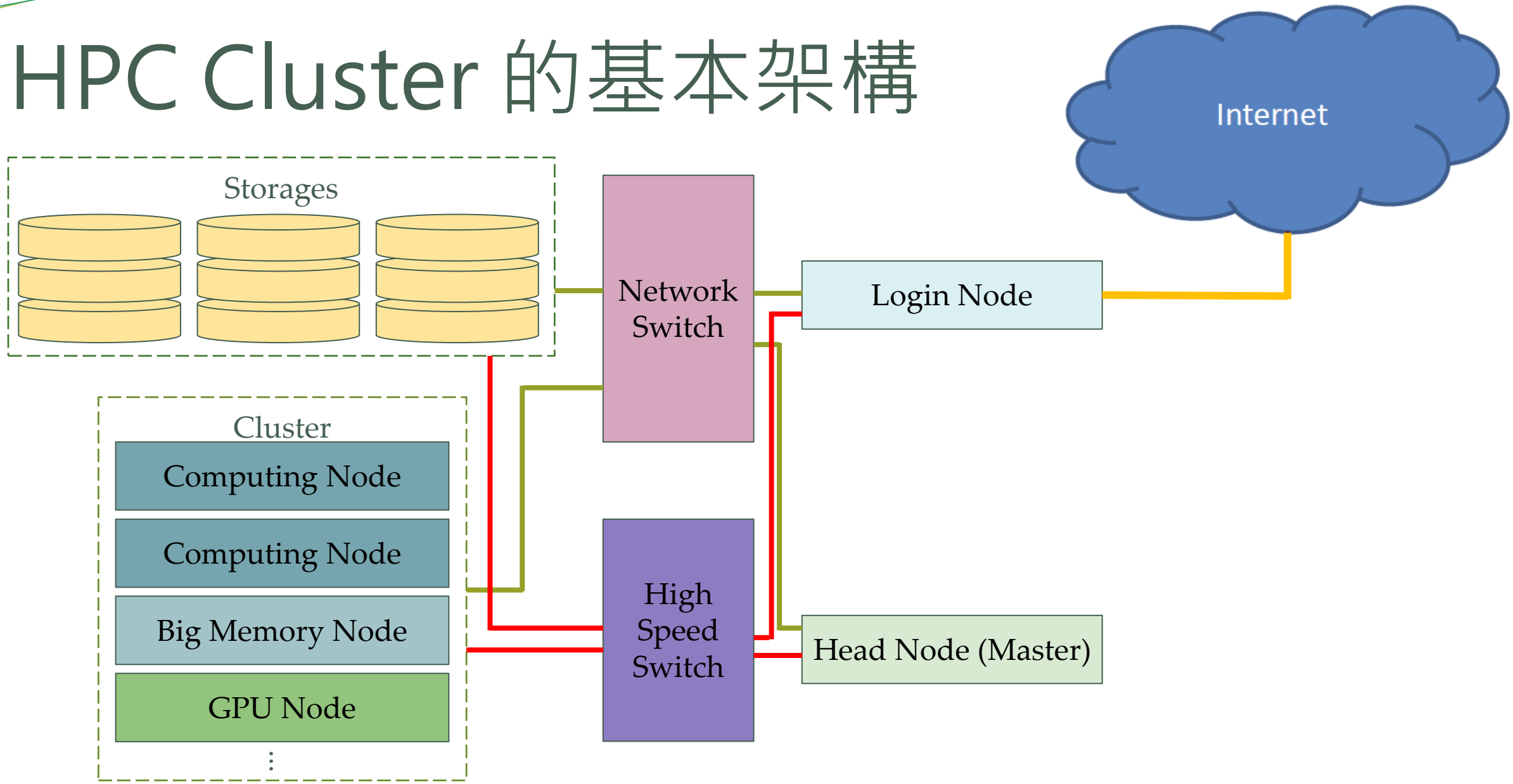
圖片來源： https://www.olcf.ornl.gov/frontier/, https://www.fujitsu.com/global/about/innovation/fugaku/

# HPC 的應用

特色

- 巨量資料分析需求
- 高效能計算需求
- 需要花很長時間處理的需求
- 大量平行分散計算需求
- 大量批次處理需求（不要人工介入）

# HPC Cluster 的基本架構



# HPC 趨勢與主流

- 高密度主機 : 1U 512 cores CPU
- GPU 加速 : many-core 8192 cores GPU
- 使用容器 : Container provides different environments
- 資料處理單元 : Offloading network functions from the CPU to the network
- 浸沒式冷卻系統 : Energy saving around 30% ~ 40%
- 雲端服務 : Pay-as-you-go

# 規劃 HPC Cluster

- 經費預算??
- 程式特性及用途，攸關 CPU 或 GPU 計算能力及主記憶體需求
- 網路交換能力
- 儲存設備速度、容量及資料備份保護（TSM）
- 作業系統及軟體（編譯器、公用函式庫、應用軟體）
- 機房空間及樓板承重（1000 kg/$m^2$）
- 機房電力、空調能源消耗（KW/BTU）
- 資訊安全（firewall, patch)

# Linux是什麼

Linux是一個開放原始碼的作業系統，目前有相當多的Linux發佈版本。主流的版本有Debian (Ubuntu, Linux Mint, Raspbian)、Red Hat (Rocky Linux, AlmaLinux, CentOS Stream, Fedora, Scientific)、Slackware、SUSE (企業版本 SLES, OpenSUSE)、Enoch (Gentoo, Chrome OS, Chromium OS)、Android


# 重要的目錄

- /dev：所有系統裝置的存取點
- /proc：系統核心、執行程序、週邊裝置狀態、網路狀態的資訊
- /sys：系統核心模組、硬體資訊
- /etc：系統主要的設定檔、帳號資料、各種服務的啟始檔
- /opt：第三方軟體的安裝位置（慣用）
- /usr：Linux 發行版內建的軟體或函式庫安裝位置
- /var：暫存檔案及系統記錄檔 **(/var/log)** 的位置
- /home：所有使用者登入時的目錄（預設家目錄）
- /root：Superuser (root) 的家目錄
- /boot：存放開機核心檔案及開機設定檔的位置

# 身為管理員，你一定要知道

- 重要的檔案
  - /etc/hosts
  - /etc/passwd
  - /etc/shadow
  - /etc/group
  - /var/log/messages
  - /var/log/secure
  - /etc/*-release
- 重要的指令
  - uname
  - env

# 身為管理員，你一定要知道

- PATH
  執行檔路徑
- HOME
  使用者的家目錄路徑
- LD_LIBRARY_PATH
  函式庫路徑

# 身為管理員，你一定要知道

- 懂得使用ssh跟基本的Linux指令
  yppasswd, ls, cp, scp, mkdir, mv, rmdir, rm –rf, man, rsync, wget, which, w, whoami, free, top, ps -ef, tar xvf, tar cvf, tar jxvf, tar zxvf, less, more...
- 熟悉文字編輯器
  Vi or Emacs

# 管理員的例行事項

- 帳號管理 (useradd, make -C /var/yp)
- 安全更新 (yum update)
- 防火牆檢查 (iptables-save)
- 檢查和新訊息和紀錄
- 檢查系統狀態 (top, ps, uptime, load, iostat...etc)
- 檔案管理 (df)
- 硬體監控 ( 事件記錄及狀態檢查 )
- 檢查排程系統 (qstat)

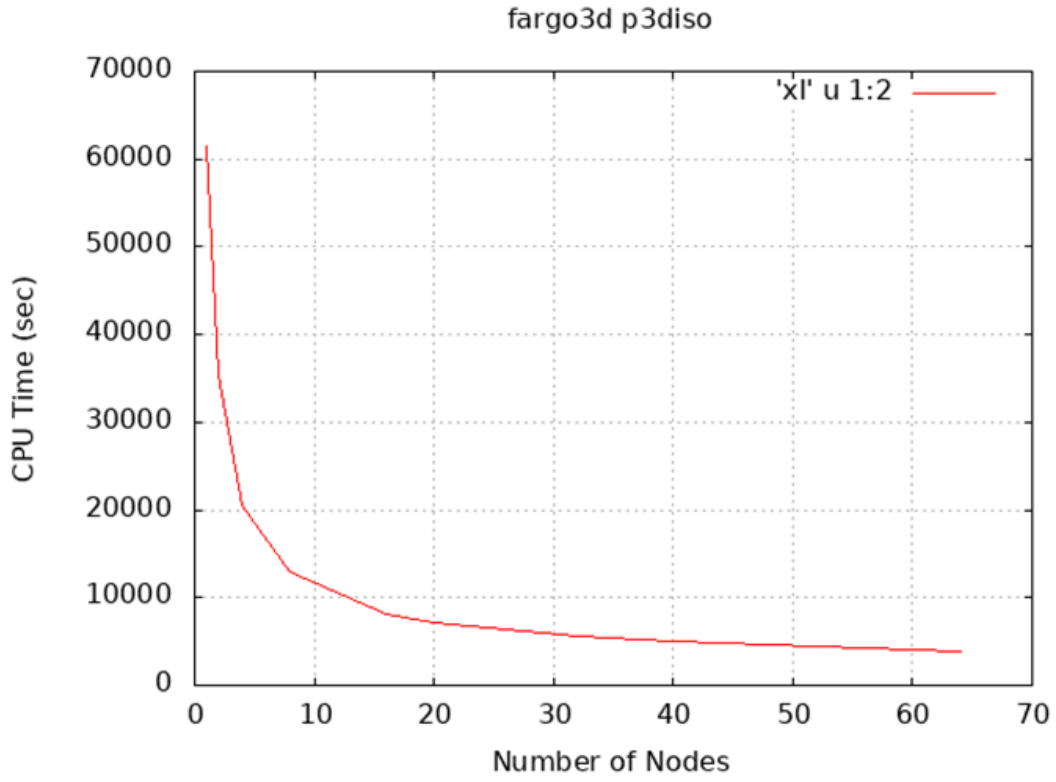
# 你需要懂得

- 如何安裝Intel編譯器
- 如何使用GNU跟Intel編譯器
- 如何分別使用GNU跟Intel的編譯器安裝OpenMPI
- 編譯並執行你自己的程式
- 如何編譯以下的函式庫:
  - FFTW3
  - BLAS
  - LAPACK
- 如何使用MKL (Math Kernel Library) 或其它函式庫

# 程式的 Benchmark

使用已知的計算實例，在不同數量的節點跟計算核心進行測試，並將測試的結果繪圖。

- 測試過程中，同時檢查每一計算步驟的結果
- 檢查最後結果的一致性
- 檢查收斂的過程
- 檢查計算的時間
- 根據“節點數”以及“計算核心”數跟計算時間繪圖



# HPL 設定與調教

$R_{peak}$ = (CPU speed in GHz) x (number of CPU cores) x (CPU instruction per cycle) x (number of CPUs per node)

| Intel Gold 6130, 16 cores, 2 CPU | | | |
|---|---|---|---|
| ISA | Base Frequency | Turbo Frequency | # of flops per cycle |
| AVX512 | 1.3 GHz | 1.9 GHz | 32 |

$$R_{peak} = 16 \times 1.9 \times 32 \times 2 = 1945.6 \text{ GFLOPS}$$

**HPL的N值**

$$N = \sqrt{\frac{total\ mem\ size(byte)}{8}} \times 0.8$$

$$N = \sqrt{\frac{64GB \times 1024 \times 1024 \times 1024}{8}} \times 0.8 \cong 82897$$

# 系統調校技巧

- BIOS 關閉 Hyper Threading
- BIOS 的 Power 設定為 Maximum Performance
- 如果系統碟是使用 HDD 設定 RAID 0，或是改使用 SSD / NVMe 不設定 RAID
- 避免將記憶體完全吃滿，關閉 swap 或避免使用 swap
- 確保使用者每次執行完程式能清除記憶體
- 確保節點對節點能跑接近網路理論值

# 程式錯誤狀況排除

- 仔細的看錯誤訊息
- 是否能夠重現程式的錯誤？
- 透過指令能夠幫助縮小範圍：
  - 確認環境變數：echo, ifort, icc, ldd, which, ompi_info
  - 確認記憶體的使用量：free -m, vmstat
  - 確認 I/O 的狀態：netstat -an, iostat, sar –n DEV 1 10, df -hT /tmp
  - 確認計算時的負載：uptime, top
  - 確認 Kernel 的訊息：dmesg | tail –n 100 > ~/klog
- 使用者回報錯誤時，一定要要求提供至少4個訊息 (4W)
  - Who：哪一個帳號？
  - Where：哪一台機器？哪一個節點？工作路徑？程式路徑？
  - When：什麼時候發生？
  - What：從輸出檔看到什麼錯誤訊息或是沒有錯誤訊息？

# 乙太網路 (Ethernet)

- 為現今最廣泛使用的區域網路類型。
  拓樸邏輯為匯流排型拓樸
  - Ethernet：10BASE
  - Fast Ethernet：100BASE
  - Gigabit Ethernet：1000BASE
  - 10Gb Ethernet：10GBASE
  - 40Gb Ethernet：40GBASE
  - 100Gb Ethernet：100GBASE
  - 200Gb Ethernet：200GBASE
- 準備向 Terabit Ethernet (TbE) 邁進

# InfiniBand

InfiniBand為一種通訊傳輸標準，具有低網路延遲及非常高的網路傳輸帶寬，原生支援遠端記憶體直接存取(Remote Direct Memory Access, RDMA)

- 10Gb IB：SDR (Retired)
- 20Gb IB：DDR (Retired)
- 40Gb IB：QDR
- 56Gb IB：FDR
- 100Gb IB：EDR
- 200Gb IB：HDR

# Subnet Managment

SM (Subnet Manager) 用於建立及管理 Infiniband 網路

- 可用 opensm 軟體管理或用有管理功能的交換器
- InfiniBand 網路特色
  - 隨插即用
  - 集中式管理
  - 1 個 SM 可以同時管理48,000 個 IB 端點

# 確認程式是否支援IB

- 將程式編譯好後用 ldd 指令確認是否使用到 IB 的 shared library

```
# ldd YOUR_BINARY
        linux-vdso.so.1 =>  (0x00007ffee718a000)
        libpthread.so.0 => /lib64/libpthread.so.0 (0x00002b0167519000)
        libm.so.6 => /lib64/libm.so.6 (0x00002b0167735000)
        libdl.so.2 => /lib64/libdl.so.2 (0x00002b0167a37000)
        libmpi_usempif08.so.0 => /opt/openmpi-1.8.8-intel17/lib/libmpi_usempif08.so.0 (0x00002b0167c3b000)
        libmpi_usempi_ignore_tkr.so.0 => /opt/openmpi-1.8.8-intel17/lib/libmpi_usempi_ignore_tkr.so.0 (0x00002b0167e7b000)
        libmpi_mpifh.so.2 => /opt/openmpi-1.8.8-intel17/lib/libmpi_mpifh.so.2 (0x00002b0168085000)
        libmpi.so.1 => /opt/openmpi-1.8.8-intel17/lib/libmpi.so.1 (0x00002b01682ee000)
        libc.so.6 => /lib64/libc.so.6 (0x00002b016889c000)
        libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00002b0168c5d000)
        /lib64/ld-linux-x86-64.so.2 (0x00002b01672f7000)
        librdmacm.so.1 => /lib64/librdmacm.so.1 (0x00002b0168e73000)
        libibverbs.so.1 => /lib64/libibverbs.so.1 (0x00002b016908a000)
        libopen-rte.so.7 => /opt/openmpi-1.8.8-intel17/lib/libopen-rte.so.7 (0x00002b016929d000)
        libtorque.so.2 => /usr/local/lib/libtorque.so.2 (0x00002b01695c2000)
 …(omit)
```

# 排除網路狀態

- 確認硬體
  - 網路線連接是否正確
  - 網卡燈號是否正常
- 由指令確認環境
  - ping
  - traceroute
  - nslookup
  - arping
  - route
  - ibstat, iblinkinfo, ibnetdiscover, ibswitches, ibhosts, ibstatus
  - smpquery