

平行與分散式程式設計

課程概述

多年以來，平行與分散式計算一直是提升運算效能的重要技術，今天仍有許多重要應用，包括人工智慧模型訓練、物理模擬、科學計算，均需要極高的運算能力，必須透過平行與分散式計算來利用大型計算叢集（Large Computing Cluster）與高效能計算（High-Performance Computing）計算平台。

本課程的目標是讓學生具備運用平行與分散式計算來解決實際問題的能力，課程名稱雖為平行與分散式程式設計，但教授的內容遠遠不止於程式設計。要知道平行與分散式計算的複雜度遠高於一般的軟體開發，開發者除了具備程式寫作、演算法設計能力之外，還需要理解應用工作的特性、計算系統中軟硬體的運作，才能進行深入的效能分析和優化，才能開發出高效能的平行與分散式程式。

因此，參與本課程的學生，務必具備程式設計、演算法、作業系統、計算機結構等基礎，否則恐怕難以學好本課程的內容，尤其是效能的分析與優化。換句話說，光是懂得平行與分散式計算的理論，或是只會寫出正確的平行與分散式程式，缺乏效能分析與優化的能力，無法實作出高效能的程式，那就缺乏實質的意義，因此本課程將會強調實作所需的能力。

在一個平行或分散式計算系統上進行實作，學生需要了解系統的：

- **應用工作的特性 (Characteristics of Application Workloads):** 例如，工作如何切割成小工作？小工作與小工作之間的相依性 (Dependency)，小工作與小工作之間的資料流動 (Data Flow)，工作量與問題大小和輸入的關係，所進行的計算種類 (整數還是浮點數)，浮點數的位元數 (FP64, FP32, FP16, FP8)，運算有哪些 (加法、乘法、超越函數)，記憶體容量的需求多大，記憶體存取的方式 (連續位址的讀寫，隨機的讀寫，只讀不寫)，計算與記憶體存取量的比例 (Arithmetic Intensity)，有多少檔案和網路的操作，工作量是否隨時間有所變化，...
- **系統架構 (System Architecture) :** 平行或分散式計算可以在多個層次實現，包括
 - 多台主機透過標準網路進行分散式計算，

- 單台主機內有多個處理機進行平行計算
 - CPU 和 GPU 協同進行平行計算
 - CPU 和 FPGA 協同進行平行計算
 - AI 加速器所採用的平行計算
- **平行與分散式程式設計 (Parallel and Distributed Programming):** 根據應用工作的特性與系統架構，選擇最合適的程式設計方法。例如：
- 在共享記憶體架構（例如多核心 CPU 晶片）撰寫平行程式常用的 OpenMP
 - 在多節點網路叢集（Network of Computers）上撰寫分散式計算程式常用的 MPI
 - CPU 和 GPU 協同計算常用的 CUDA 程式設計
 - CPU 和 FPGA 協同計算的程式設計流程
 - 設計加速器所使用的高階合成（High-Level Synthesis）技術
- **效能優化、程式庫、工具 (Performance Optimization, Libraries, and Tools):** 平行與分散式計算講求效能，如何蒐集與效能相關的資訊並加以分析？如何根據實際表現優化程式庫的設定或使用方式？如何建構效能模型來探討效能與軟硬體組態的關係？如何運用效能剖析與追蹤工具（Profiling and Tracing Tools）的協助？

本課程為大學部高年級與研究所選修課程，除了課程講授與實驗之外，也要求學生自行研讀補充教材，以及尋找適合期末專題的研究題材。本課程將採用「做中學」(Problem/Project-based Learning) 的教學思維，引導學生在解決實務問題、進行期末專題的同時，探討平行與分散式程式設計的內涵。除了於課堂上講授知識與技能之外，學生必須在課堂之外學習實務與進階的技能。課程中將包含 5 次實作型作業以及期末專題，成績將根據實作程式的效能而定。

課程目標

本課程的目標是讓學生具備運用平行與分散式計算來解決實際問題的能力，課程名稱雖為平行與分散式程式設計，但教授的內容遠遠不止於程式設計。要知道平行與分散式計算的複雜度遠高於一般的軟體開發，開發者除了具備程式寫作、演算法設計能力之外，還需要理解應用工作的特性、計算系統中軟硬體的

運作，才能進行深入的效能分析和優化，才能開發出高效能的平行與分散式程式。

本課程的目標在於讓修課同學：

- 理解應用工作的特性與平行/分散式計算的關聯性
- 理解各種平行/分散系統架構、分析系統中軟硬體的互動
- 學習在各種平行/分散系統架構上發展平行化程式及演算法
- 使用效能工具分析與優化平行/分散式程式的效能
- 培養尋找研究題材與進行研究之專業能力

課程要求

本課程為大學部高年級與研究所選修課程，除了課程講授之外，也要求學生自行研讀教材，進行實驗，以及設計期末專題。學生需具備下列課程基礎：

- 程式設計
- 作業系統
- 計算機結構

預期每週課前與課後學習時數

- 每週上課前以 1 至 2 小時預習指定的預先閱讀材料（如果有的話）。
- 每週上課後以 1 至 2 小時複習上課的內容，於線上或下次上課時討論。
- 預期每 2 至 3 週至少 6 個小時完成一個作業。
- 預期以至少 12 個小時完成期末專題。

指定閱讀

上課時提供。

面談時間

另約時間。

評量方式

上課表現20%，作業50%，期末報告30%（可能視實際狀況調整）

授課時可能的調整方式

- 學生預先請假核准，可能以錄影提供請假學生使用。
- 學生因特殊理由導致作業遲繳或無法參與課堂討論、期末報告時，可與教師與助教協商處理方式。

課程進度

週次	日期	單元主題
1	2/27	(國定假日, 請欲加選者填寫問卷)
2	3/06	Overview and quiz
3	3/13	Shared memory systems and OpenMP programming
4	3/20	Performance analysis and optimization
5	3/27	Distributed memory systems and MPI programming
6	4/03	(國定假日)
7	4/10	GPU and CUDA programming
8	4/17	High-performance computing and AI accelerators
9	4/24	FPGA and high-level synthesis
10	5/01	(國定假日)
11	5/08	AI-assisted parallel programming
12	5/15	Review of research papers
13	5/22	Final project proposals
14	5/29	Advanced case studies/Guest Speakers
15	6/05	Advanced case studies/Guest Speakers
16	6/12	Final project report