

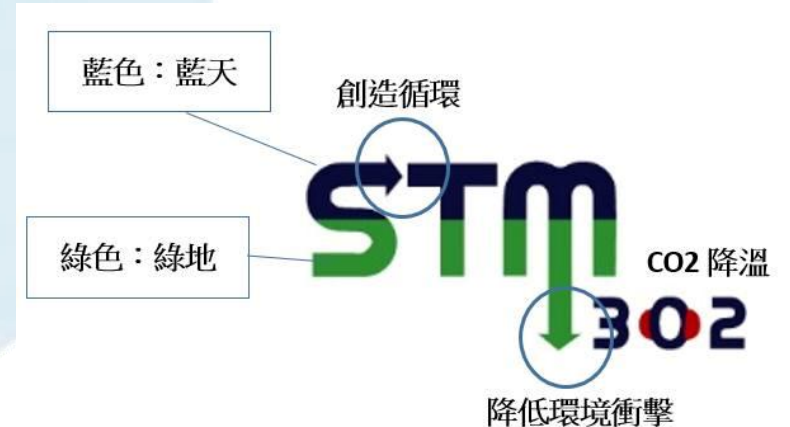
請尊重著作權！

本投影片內容所有權為國立臺灣科技大學，
請勿任意轉傳，或轉貼。

About me



- 現職
 - 臺科大教授，永續能源研究基金會秘書長，上市櫃獨立董事
- 學歷
 - 德州理工大學工業工程碩士，博士
- 專長
 - 永續發展，循環經濟，碳風險管理，數位轉型與智慧製造
- 經歷
 - 科技部自然科學及永續研究發展司永續學門共同召集人
 - 台科大管理學院副院長，工業管理系系主任
 - 中原大學電資學院副院長，工業與系統工程系系主任
- 得獎
 - 2020~2025 全世界2%科學家
 - 國立臺灣科技大學傑出研究獎2次
 - 國立臺灣科技大學教學優良獎1次
 - 中國工業工程學會學術貢獻獎1次
 - 國立臺灣科技大學優良研究獎1次
 - 中原大學傑出研究獎
 - 台達文教基金會企業倫理獎
 - 科技部延攬優秀人才獎勵

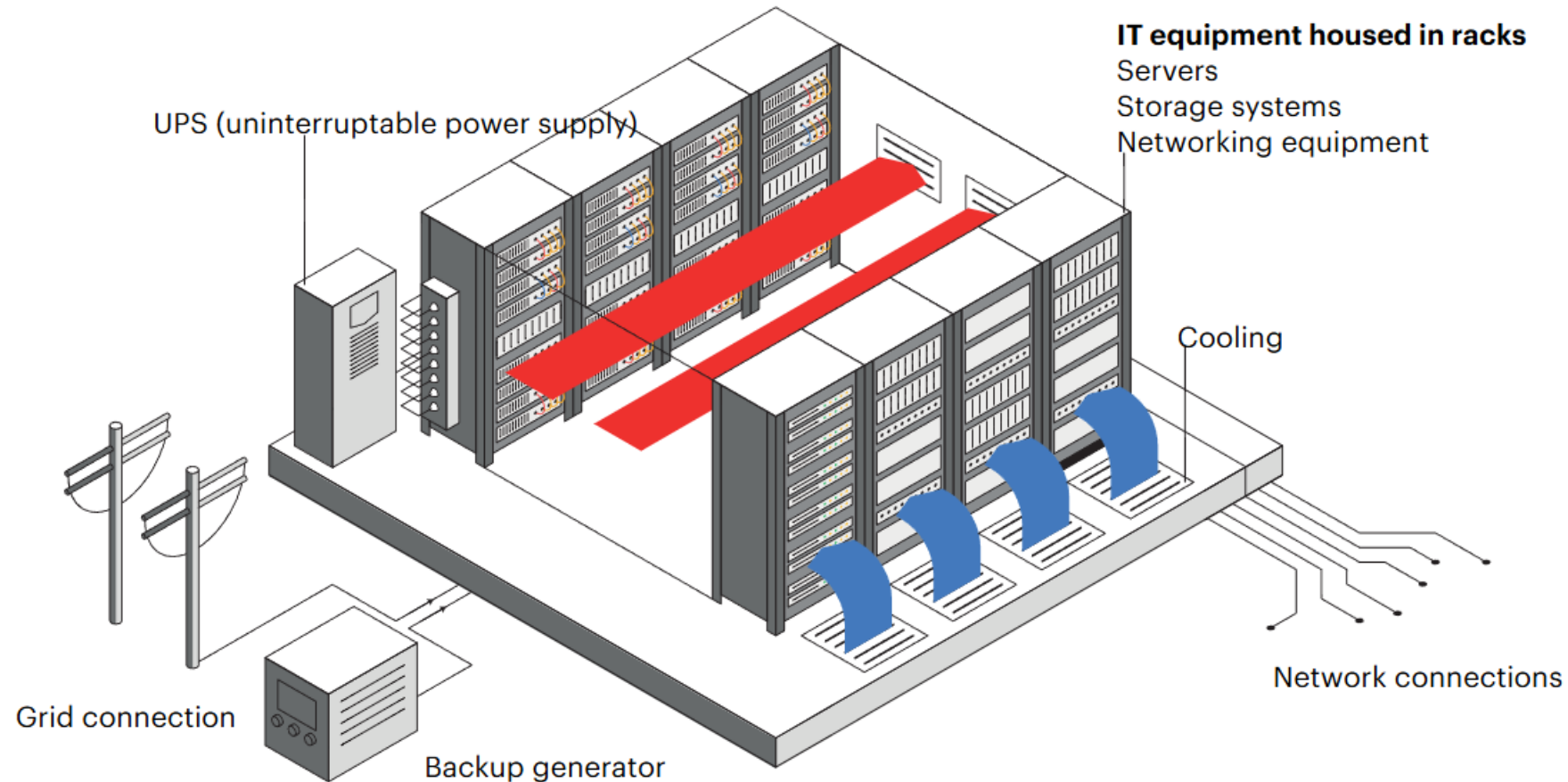




自願性綠色軟體標章
綠色軟體開發與國際市場新契機

郭財吉教授/秘書長
永續科技管理研究室
國立臺灣科技大學工業管理系
台灣永續能源研究基金會

AI對能源的需求



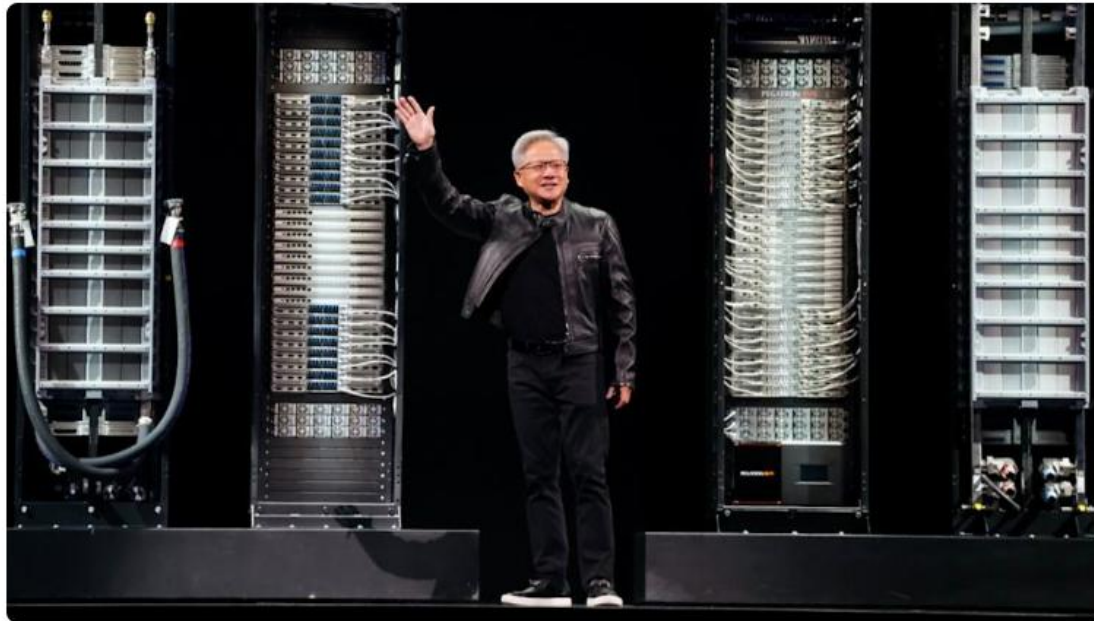
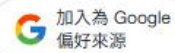
LLM耗能

- 使用十次大型語言模型(如GPT-4等級)的推論請求，平均消耗約0.03-0.05瓩小時 (kWh，度電)，可以用筆電60分鐘。

【吃電怪獸3-1】AI答題用電量驚人！2機櫃輝達GB300電力負載就超過整棟台電大樓

吳馥馨

2025年12月20日



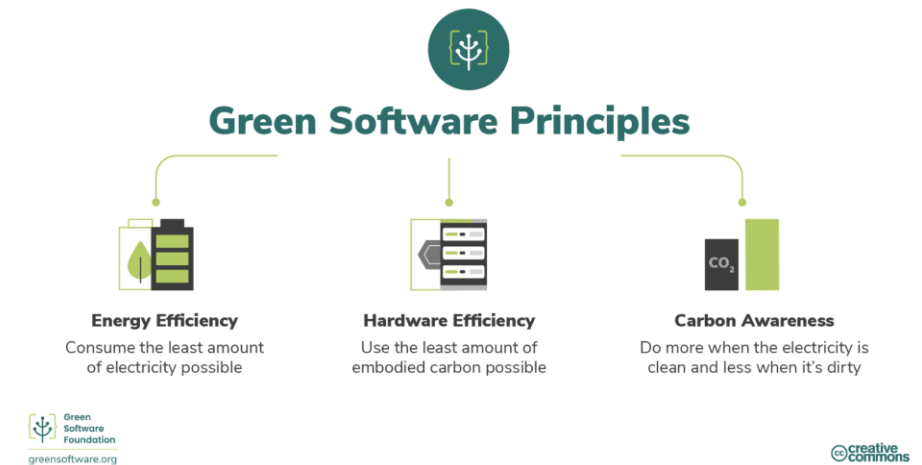
2個機櫃的GB300要價1億元，成本考量，初期裝設量還不至太多太快。

但
台電董事長曾文生則具像化說明：以輝達GB300伺服器為例，1個機櫃的電力負載就1.4MW，2個機櫃就高達2.8MW，用電量遠遠高於整棟台電大樓的2.5MW

資料來源：吳馥馨 2025年12月20日

綠色軟體基金會 (Green Software Foundation, GSF)

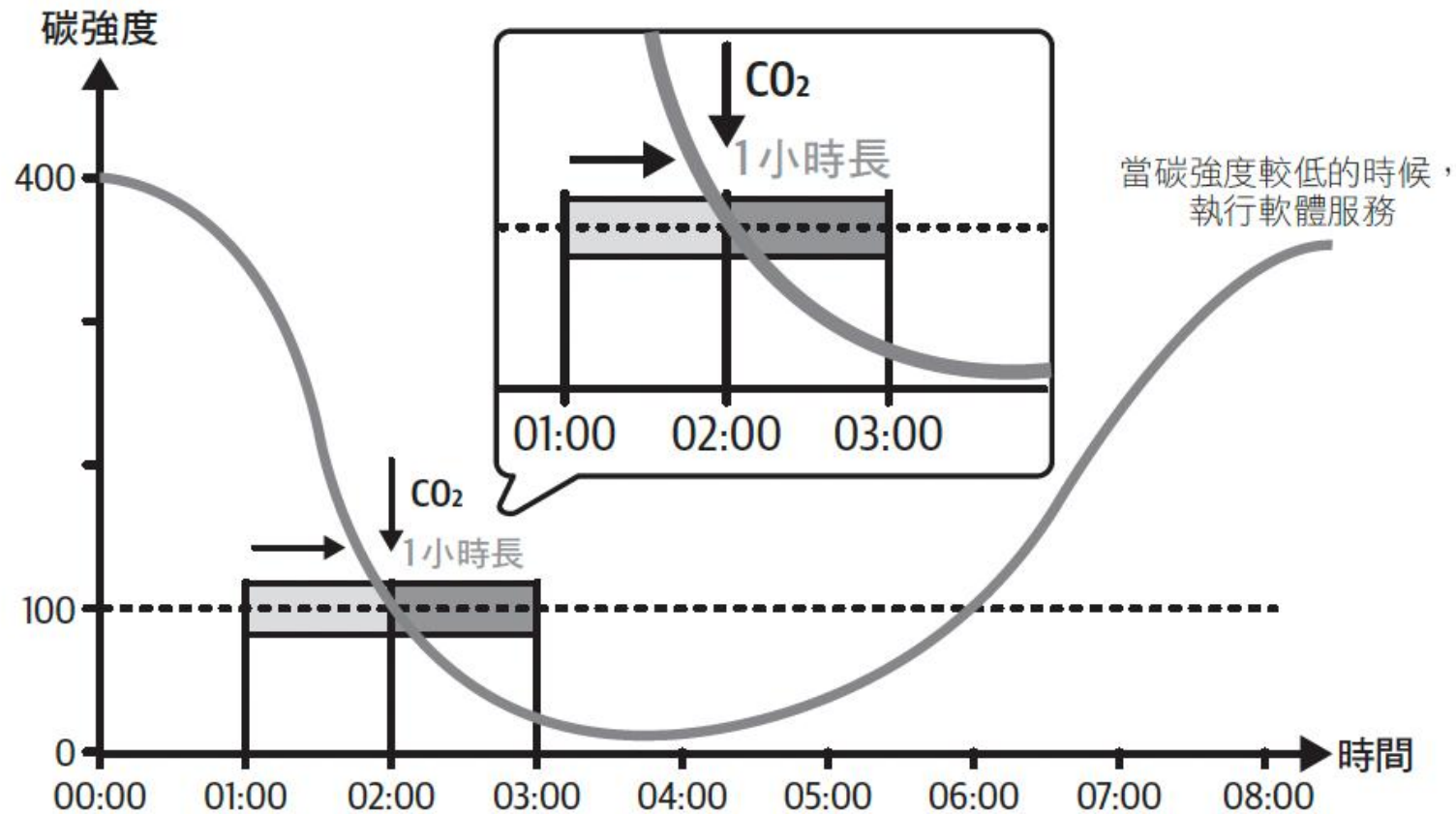
- 由Linux基金會等科技巨頭支持的倡議組織。
- 推動**綠色軟體**原則 (Principles of Sustainable Software Engineering)，例如：**碳效率**、**能源效率**、**碳感知**、**硬體效率**等。
- 發展**軟體碳強度 (Software Carbon Intensity, SCI) 規範**，作為衡量軟體永續性的指標，未來可能演變成一種可比較的「標籤化」
- 只有三種行動可以減少軟體的碳排放：
 - 使用更少的實體資源
 - 使用更少的能源
 - 更智慧地使用能源



碳感知

- 「碳感知」代表著從被動的「節能減碳」走向主動的「智慧追綠」。透過軟體演算法與即時數據，讓科技產品的運行與地球的生態律動（如日出、風吹）同步，是達成淨零碳排（Net Zero）不可或缺的智慧化手段。
 - **預測碳強度**：利用 AI 和機器學習模型預測未來 24 小時甚至更長時間的電網碳強度。
 - **設定排程**：軟體排程器（Scheduler）根據預測結果，自動將工作負載安排在碳強度最低的時段開始。

時間移轉~環保時間



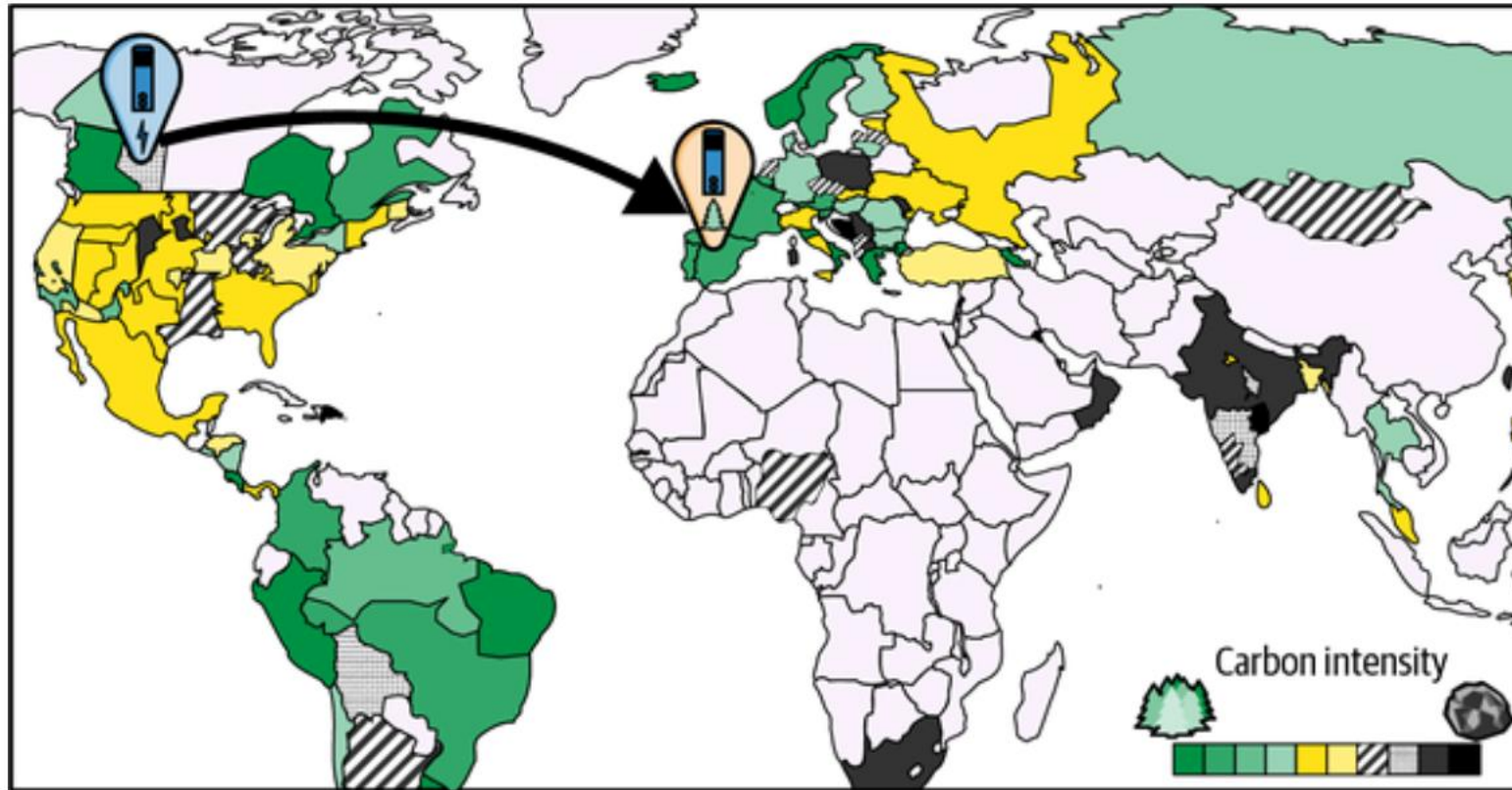
用時間的轉變來實現碳減排

機器學習模型的訓練階段，於一天中最環保的時間

從時移中獲得巨大收益的場景是批次處理。批次作業是一種工作負載，它將多個任務分組並一個接一個地執行，無需即時使用者輸入。例如，軟體更新和系統備份幾乎可以在給定時期內的任何時間發生。

資料來源：O'Reilly Building Green Software <https://www.oreilly.com/library/view/building-green-software/9781098150617/>

空間轉移



空間轉移可以是減少整體碳排放的富有成效的做法。

Google Cloud Platform (GCP) 在亞洲位置部署了一個長期運行的應用程式，並且如果延遲（或任何其他問題）不是問題，則將其移動到自然具有低碳能源的地區來源可以大大減少應用程式的碳足跡。

資料來源：O'Reilly Building Green Software <https://www.oreilly.com/library/view/building-green-software/9781098150617/>

■規範說明：

- SCI是一種分數型指標。分數越低，代表軟體對環境的影響越小，鼓勵持續降低分數的永續目標。
- 著重說明使用者和開發人員瞭解如何改進軟體以減少或避免產生排放。
- ISO 21031聚焦減少或避免實際排放的產生，因此SCI分數僅能透過實際消除碳排來源來降低，例如使用較少的硬體資源、降低能源消耗、使用低碳或再生能源等。
- SCI分數無法透過碳中和(Carbon Neutrality)、碳抵換(Carbon Offset)等方法降低。

■適用場景：

- SCI可在個人電腦、私有資料中心，或是超大規模的公有雲平台的環境上運行，包含大型分散式雲端系統(distributed cloud system)、單機版開源函式庫(small monolithic open source library)、本地應用程式(on-premise application)、無伺服器架構(Serverless Functions)等。

軟體在CPU上運行1小時所產生碳排放

$$SCI = ((E * I) + M) \text{ per } R$$

$$SCI_{CPU} = ((E_{CPU} * I) + TE_{CPU} * TS_{CPU} * RS_{CPU}) \text{ per } R_{1\text{小時}}$$

以使用時間計算CPU消耗電量¹
並以軟體使用CPU%作為電量占比

CPU的隱含碳排
使用產品碳足跡

一小時軟體碳強度

資源占比

軟體使用CPU%作為資源占比

1. 台灣113年電力排碳係數0.474
2. 每小時電力碳排

軟體運行時間vsCPU使用年限²占比

資料來源¹² : IntelCPU

<https://www.intel.com.tw/content/www/tw/zh/products/sku/241947/intel-core-9-processor-270h-24m-cache-up-to-5-80-ghz/specifications.html>

<https://www.intel.cn/content/www/cn/zh/support/articles/000024255/processors.html>

$$SCI = ((E * I) + M) \text{ per } R$$

■計算方法

$$\square SCI = C \text{ per } R$$

排碳總量C，每個功能單元R，可以是：使用者、機器、時間（分）、資料量等

$$\square SCI = (O + M) \text{ per } R$$

軟體應用程式的操作過程中的碳排放量O，硬體裝置建立和處置過程中的碳排放量M (隱含碳排)

$$\square O = (E * I)$$

功能單元工作消耗的電量E (kWh為單位)，電力的碳強度I (gCO₂eq/kWh為單位)

$$\square M = TE * TS * RS$$

TE=總體隱含排放，TS=時間份額(指硬體總使用壽命為軟體保留的使用份額)，RS = 資源份額(指硬體總可用資源中為軟體保留的使用份額)

$$\square M = TE * (TiR/EL) * (RR/ToR)$$

TiR=保留時間(硬體為軟體保留使用的時間長度) EL=預期壽命(設備預計安裝使用的時間)

RR=保留資源(為軟體保留使用的資源數量) ToR=資源總量(可用的資源總數)

$$\square SCI = ((E * I) + TE * (TiR/EL) * (RR/ToR)) \text{ per } R$$

資料來源：ISO/IEC (2024). Information technology — Software Carbon Intensity (SCI) specification (ISO/IEC 21031:2024). <https://www.iso.org/standard/86612.html>

綠色軟體架構

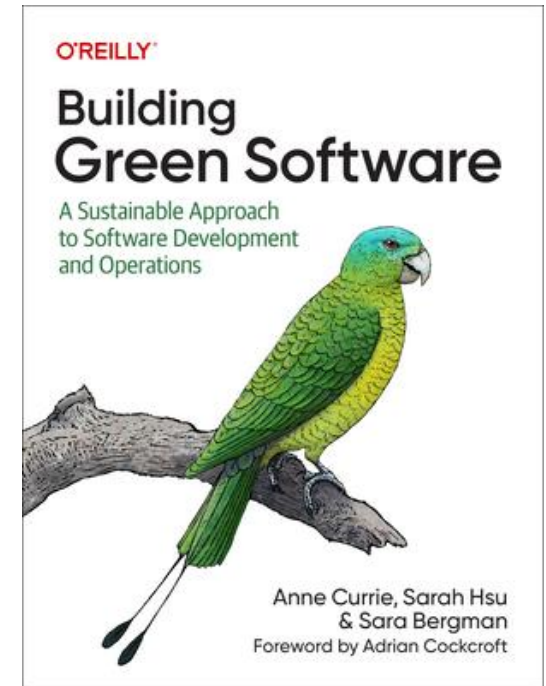
Green Software Framework

Impact area	Green software practices
Green Software development lifecycle	在軟體開發生命週期中採用節能與綠色實踐， 從選擇平台、程式語言，到設計軟體架構及 DevOps ，以節省能源、減少排放，開發碳效率高的軟體
Green UI/UX	透過 有效的無障礙設計提升用戶友好的數位體驗 - 簡化導航減少螢幕使用時間以降低排放；選擇螢幕顏色、評估螢幕處理能力並壓縮內容和圖像來優化性能
Green AI	根據使用情境的重要性， 評估 AI/ML 模型之能耗與準確度之間的取捨 。並將現有模型透過轉移學習重新用於不同任務，可進一步減少能源與時間消耗，從而降低排放
Green Data	在數據生命週期中，通過 降低儲存浪費、數據壓縮、有效利用網絡與數據傳輸 ，專注於高效的數據處理，減少系統中的無用數據，從而降低碳排放
Green Cloud & data centers	鼓勵遷移到雲端 ，選擇合適的託管方案及在雲端上進行綠色軟體開發，以提升硬體與能源效率；進一步 評估邊緣運算的實施 ，在設備或終端用戶附近儲存與使用數據
Green Infrastructure	通過考慮使用過程排放以及與製造和產品壽命終點相關的生命週期排放，推動減少 IT 基礎設施的環境影響。 鼓勵負責任的採購和產品壽命終點管理 等措施

資料來源：[Uniting technology and sustainability | Accenture](#)

建構綠色軟體

- Building Green Software: A Sustainable Approach to Software Development and Operations
 - 能源轉型將如何改變本地和雲端的托管方式，以及您的公司如何做好準備
 - 可持續軟體開發的基本架構原則及其應用
 - 如何確定系統中需要改變的部分
 - 延長硬體壽命的概念以及軟體在其中的角色



軟體溫室氣體排放之計算步驟

功能單元 (計算目標)

- 1.API call/request API 呼叫 / 請求
- 2.Benchmark 基準
- 3.User 使用者
- 4.Machine 機器
- 5.Minute/time unit 分鐘 / 時間單位
- 6.Device 裝置
- 7.Physical site 實體站點
- 8.Data volume 資料量
- 9.Batch/Scheduled Job 批次 / 排程作業
- 10.Transaction 交易
- 11.Database read/write 資料庫讀寫



軟體邊界 (計算邊界)

compute resources 運算資源	testing 測試
storage 儲存設備	training ML models 機器學習模型訓練
networking equipment 網路設備	operations 系統運維
memory 記憶體	backup 備份
monitoring 系統監控	resources to support redundancy 系統備援的資源
idle machines 閒置機器	resources to support failover 故障切換的資源
logging 日誌紀錄	End user devices 終端使用者裝置
scanning 掃描	IoT devices 物聯網裝置
build and deploy pipelines 建置與部署流程	Edge devices 邊緣運算裝置

資料來源：ISO/IEC (2024). Information technology — Software Carbon Intensity (SCI) specification (ISO/IEC 21031:2024). <https://www.iso.org/standard/86612.html>

SCI計算

數據設定（優化前）：

1. 能耗 (E)：處理 100 萬筆訂單，雲端伺服器（含 CPU、記憶體、資料庫讀寫）總共消耗了 5,000 度電 (kWh)。
2. 電網碳強度 (I)：該資料中心位於一個主要依賴**燃煤**發電的地區，當時的電網碳強度為 500 gCO₂e/kWh。
3. 硬體隱含碳 (M)：根據雲端供應商提供的**生命周期評估 (LCA)** 數據，這群伺服器在處理這批訂單的時間內，依比例分攤到的硬體製造與報廢之碳排放為 250,000 gCO₂e。

💡 優化前的 SCI 計算： $SCI = (5,000 * 500) + 250,000 = 2,750,000 \text{gCO}_2\text{e}$
也就是說，每處理 100 萬筆訂單，軟體強度為 2.75 噸碳排放

策略一：降低能耗（降低 E）

策略二：碳感知調度（降低 I）

策略三：提高硬體效率（降低 M）

納入綠色採購條件

綠色本質

- 源頭減量
- 綠色設計(低碳排、使用再生能源)
- 節能高效

公信力

- 核發單位
- 計算標準依據

市場可行性

- 產品數量、市場需求

標章管理機制

- 相關法規、運作管理機制

自願性綠色軟體標章設計符合項目

綠色軟體設計

- 能源效率
- 硬體效率
- 碳意識

市場可行性

- 軟體產品
- 綠色競爭力
- SCI計算

標章管理機制

- 標章管理辦法
- 審議、驗證、管理機制



自願性綠色軟體標章
綠色軟體開發與國際市場新契機

郭財吉教授/秘書長
永續科技管理研究室
國立臺灣科技大學工業管理系
台灣永續能源研究基金會