

使用 sched-ext 自訂 CPU 排程器

Customize CPU scheduler with sched_ext

組員： 許育瑋 指導教授： 黃敬群 教授, 涂嘉恒 教授 國立成功大學 資訊工程學系

研究動機與目的

Linux 預設排程器 **EEVDF** 在多數情境表現良好，但可調參數有限，難以針對特定場景最佳化。nice 值受公平性限制效果有限；cgroup 的最小分類單位是 **process**，無法讓同一 process 內的不同 thread 享有不同資源權限。

Linux v6.12 正式合併 **sched_ext**，能以 eBPF 程式修改排程行為。受惠於 sched_ext 與 eBPF 的防護機制，排程器出錯**也不會造成 kernel panic**，更不需深入理解核心排程邏輯，大幅提升開發與實驗效率。

本專案提供一個基於 sched_ext 的**排程器框架–scx_teddy**，目標是**降低特定任務的延遲**。使用者只需了解 workload，依 README 蒐集資料、訓練機器學習模型並設定參數，即可優化目標任務延遲；即使不懂 workload 性質，框架也提供**預設模型與參數**供一般使用者直接套用。

為什麼需要機器學習

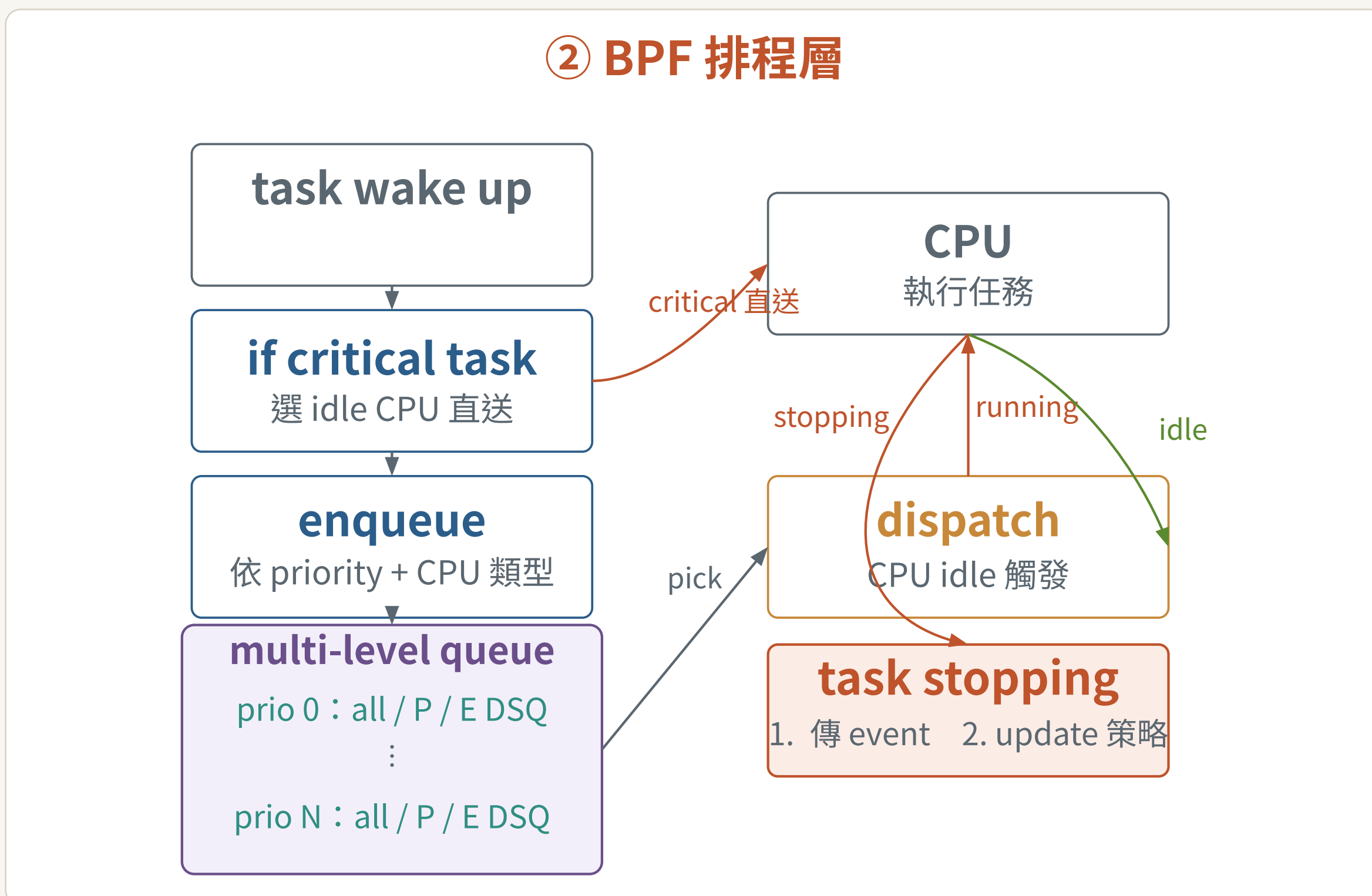
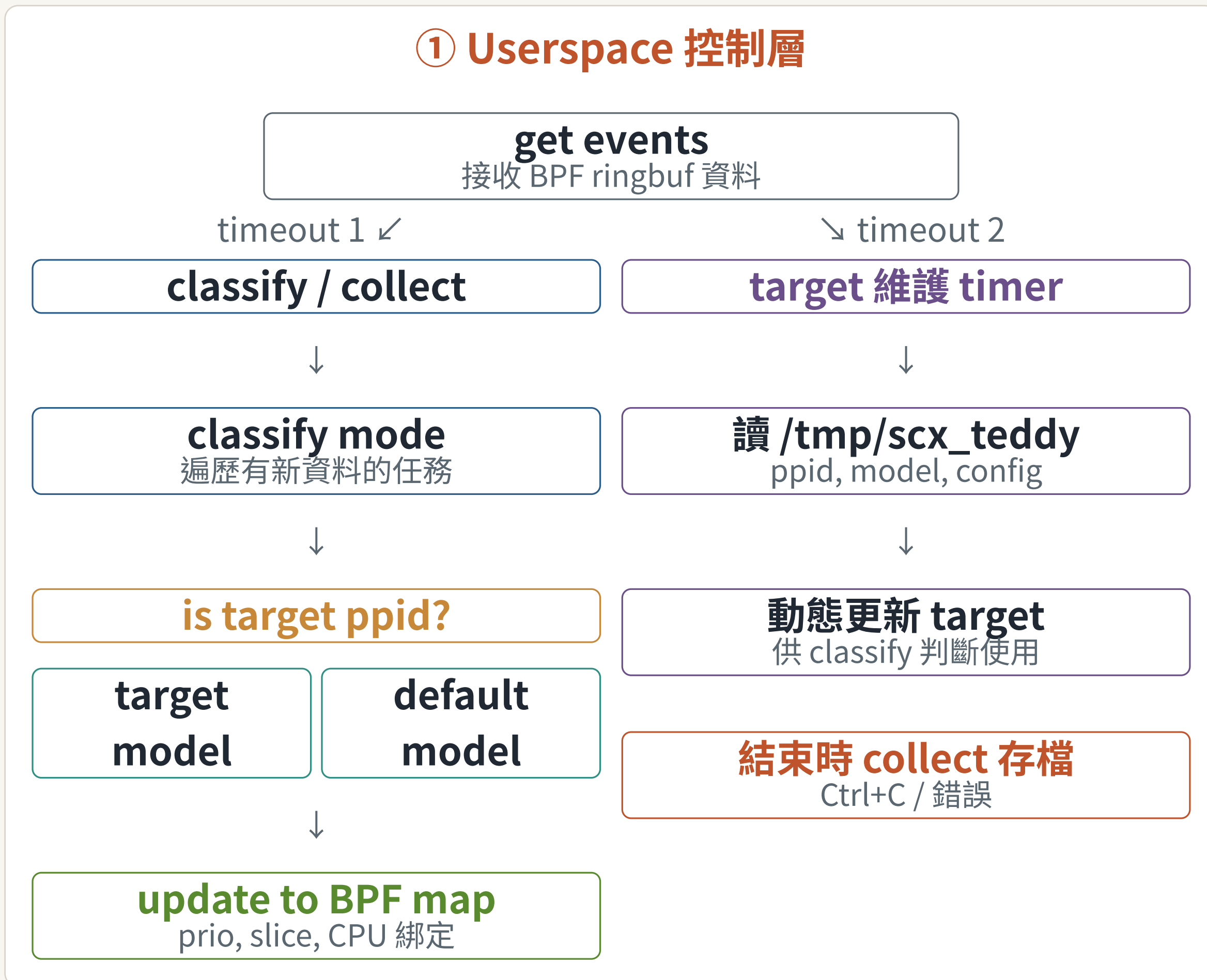
若能完整掌握目標 workload 的行為（即握有原始碼），理論上可用**規則式排程**達成最佳化。但現實中往往拿不到原始碼，或原始碼過於龐大而無法理解其行為。

此時**統計模型**便派上用場：將模型擬合到蒐集的資料，即可大致預測任務行為 — 這就是機器學習。本專案採用 **K-means** 作為分類方法，透過迭代計算各 cluster 中點達成分類，並假設**同類型任務的 feature 長時間會收斂到相近數值**。

對比排程器

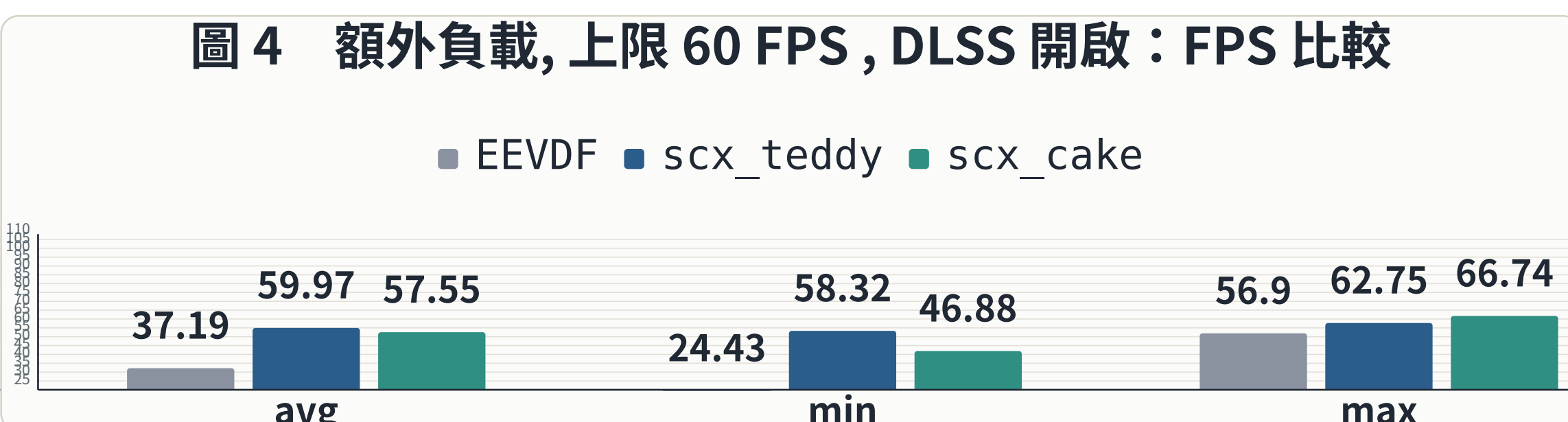
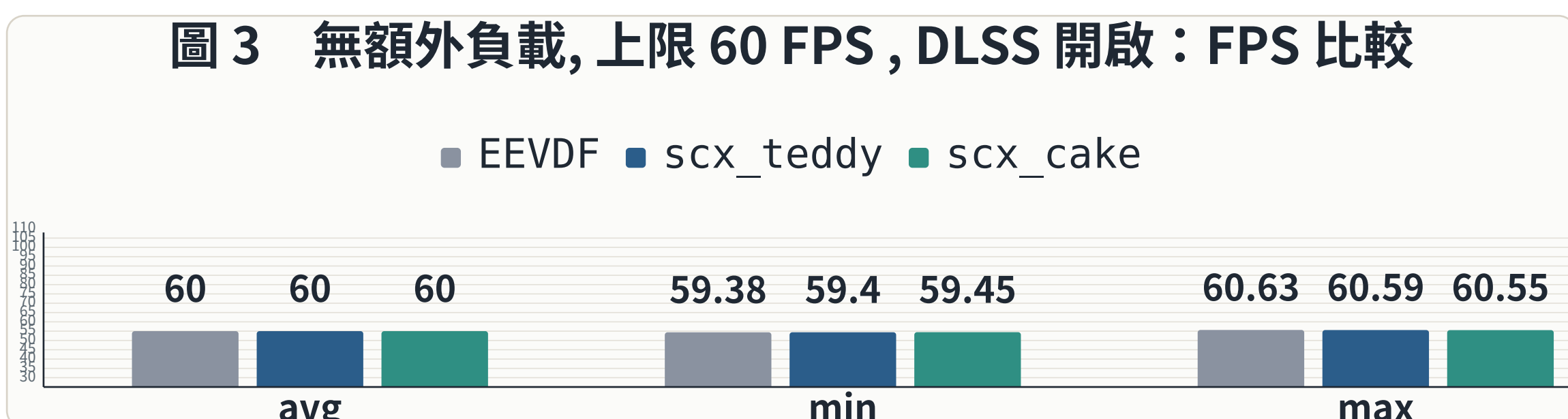
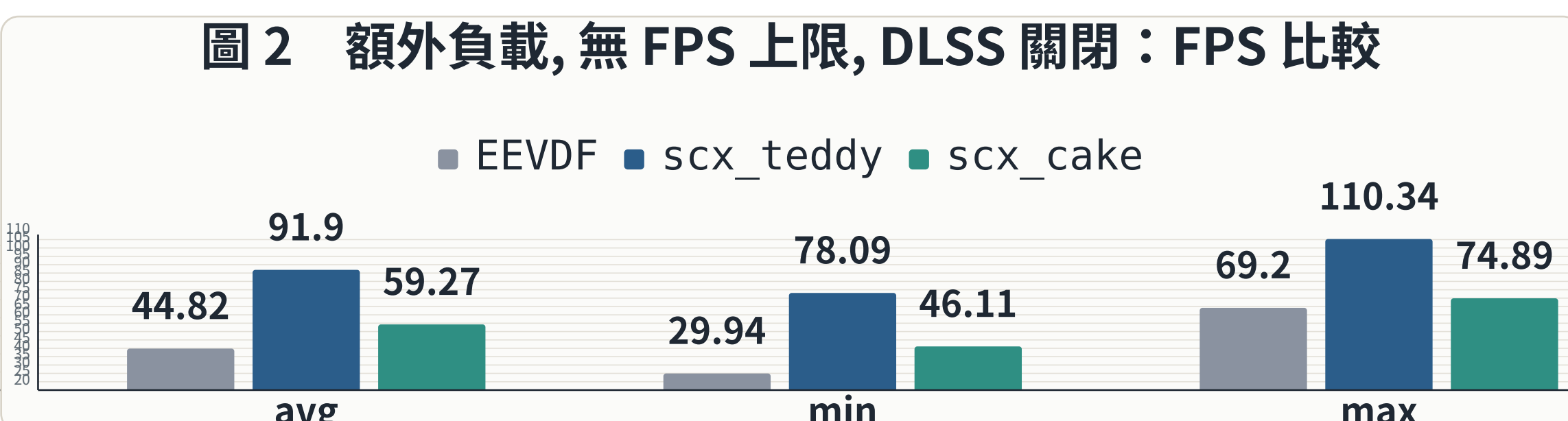
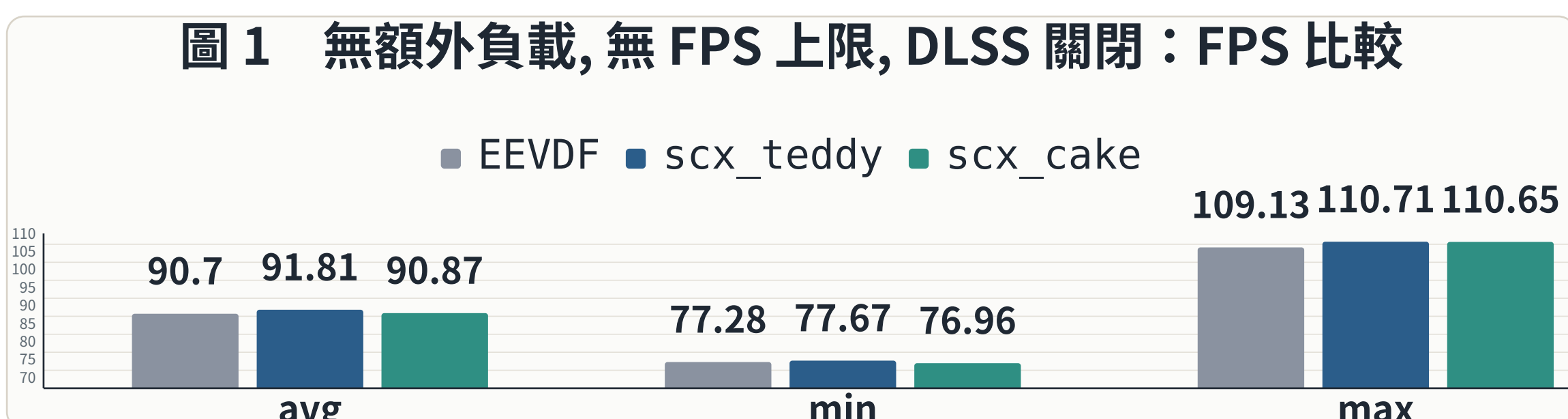
- EEVDF**：Linux 預設排程器，重視公平性。作為未調整的基準線。
- scx_cake**：專為遊戲設計的 sched_ext 排程器，本專案的靈感來源（如 Steam 遊戲偵測、不重公平而重遊戲效能的策略）。本框架將此概念一般化，透過機器學習適配任意 workload。

系統架構



成果展示 Results

以 **Cyberpunk 2077** 內建 benchmark 為測試環境，比較本框架（scx_teddy）、scx_cake（遊戲訂製排程器）與預設 EEVDF。



結論

本框架讓使用者能更靈活地針對特定任務**降低延遲**。只要能充分分析任務性質，便可獲得更好的效果。

在 Cyberpunk 2077 benchmark 中，無負載時表現與專用排程器相當，**有負載時顯著勝出**，兼顧 FPS 上限與穩定度。

未來展望

- 針對負載做更完整的分析
- P-core deadline 機制**（預設綁 P-core 的任務 10ms 未取得 CPU 即可改用 E-core）開放為使用者可調參數
- Feature 工程優化：
 - 導入機制辨識 **IO 事件來源**（如等待 GPU）
 - 移除 runtime_ms / avg_sleep_ms，避免過大值扭曲數據
 - 導入 **EWMA**，因應任務不同階段的性質差異（如遊戲標題畫面 vs. 實際遊玩）
- 介面開放**替換自訂分類演算法**，不綁定於 K-means

Demo



QR Code
(Demo 影片)