

深度學習

專題編號：112-CSIE-S016

執行期限：111 年第 1 學期至 112 年第 1 學期

指導教授：劉傳銘

專題參與人員：109590450 張如儀

一、摘要

本研究旨在提升固態硬碟 (SSD) 緩存管理效能，特別針對垃圾回收 (GC) 進行改進。主要使用 Python 以及 Tensorflow，建立了兩種不同的 SSD 緩存策略，分別是傳統的 LRU 方式和基於深度學習 (Deep Q-Learning) 的自適應緩存。透過這兩種方法，分析並比較其緩存命中率 (hits) 和未命中率 (miss)，以確定深度學習是否提供更有效的 SSD 緩存管理。這項研究的目標為提高儲存系統的性能，優化 SSD 的運作，同時提升資料存取的效率。研究結果將有助於確定使用 Deep Q-Learning 是否能夠更有效地改善 SSD Cache 的 GC 效能，以應對大規模數據存儲的挑戰。

關鍵詞：SSD、Deep Q learning、深度學習

二、緣由與目的

做深度學習的緣由隨著數據量增加和存儲系統性能要求提高，SSD 已成為許多應用的首選存儲設備。然而，SSD 在長時間連續寫入或頻繁更新時會產生大量垃圾數據，導致有效數據存儲和讀取困難，造成性能下降。

本研究旨在利用深度學習的 Deep Q-Learning 改善 SSD Cache 中垃圾回收 (GC) 的效能。使用 Python 建立兩種策略：一是基於最近最少使用 (LRU) 的緩存機制，另一是使用 Deep Q-Learning 算法進行緩存篩選。

通過比較兩種策略的 hits 和 miss 數量，能夠評估深度學習方法在 SSD Cache 中 GC 效能上是否有優勢。

三、使用技術方法

(一) Q-Learning (QL)

Q-learning 是強化學習的一種方法，主要是透過記錄學習過的策略，來告訴智能體 (Agent)，什麼情況下要對應採取什麼行動 (Action) 會得到最大的講獎勵 (Reward)。Q-learning 會將對應行動的獎勵值存在一個 Q 表中 (Q-table)。

Q-Learning 為 value-based 的強化學習算法，在 state-action value function of $Q(s,a)$ 表示在狀態 s 中選擇動作 a 並在未來採取的所有動作中表現最佳的未來最大獎勵。

$$Q(st,at) = \max Rt + I \quad (1)$$

將 $Q(s,a)$ 視為環境中每個狀態 s 的最佳可能動作 a ，達在最佳可能的 state-action value。可以透過計算 $Q(s,a)$ 和 $Q(s',a')$ 的值，根據 policy(2) 來選擇具有最佳 Q-value 的動作。

$$\pi(s) = \operatorname{argmax}_a Q(s, a) \quad (2)$$

這裡的 π 表示策略函數，即在給定狀態下如何選擇特定動作的規則。為了獲得 Q 函數，考慮一個轉換 $\langle s, a, r, s' \rangle$ 。狀態 s 和動作 a 的 Q 值根據下一個狀態 s' 的 Q 值在 (3) 中給出。

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a') \quad (3)$$

通過不斷迭代更新 Q 值，Q-Learning 能夠學習最佳策略，以在不同狀態下選擇最佳行動，以最大化總體獎勵。

(二) Deep Q-Learning (DQL)

Deep Q-Learning (DQL) 是

Q-Learning 的擴展，專為處理具有大量狀態空間的情況而設計。在傳統的 Q-Learning 中，使用 Q-Table 來存儲每個狀態-行動對的估計值，但當狀態空間變得龐大時，這種方法變得不切實際。DQL 採用了深度學習技術，特別是卷積神經網絡 (CNN)，來取代 Q-Table，能夠處理比較大和更複雜的狀態空間。

在 DQL 中，輸入為狀態 (state)，而輸出為每個可能動作的 Q-value，即 $Q(s, a)$ 。這是通過使用 CNN 模型進行逼真的近似估算而實現。DQL 模型訓練的目標是最小化 Q-value 的均方誤差損失，使其逼近 Bellman 方程中的目標值。

DQL 的優勢在於能夠處理具有高度複雜性和大規模狀態空間的問題，如 Atari 遊戲等。通過使用深度學習模型，DQL 可以自動提取和學習有關狀態的特徵，而不需要手工設計特徵。這使得 DQL 成為處理各種強化學習應用程序的有效工具。

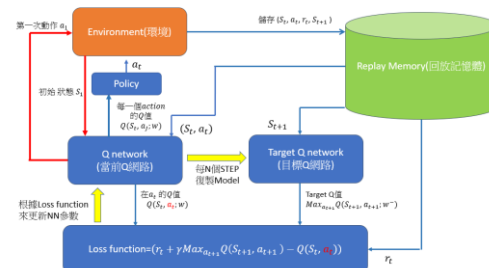
DQL 的訓練過程通常涉及到使用經驗回放 (Experience Replay) 和固定目標網絡 (Fixed Target Network) 等技術，以穩定和加速訓練過程。經驗回放允許模型從以前的經驗中學習，而固定目標網絡則用於穩定目標 Q-value 的估算，以減少訓練過程中的震盪。總之，Deep Q-Learning 通過整合深度學習技術，尤其是卷積神經網絡，使其能夠應對複雜和大規模狀態空間的強化學習問題，並透過訓練以優化 Q-value，以實現更好的策略和性能。

(三) LRU (Least Recently Used)

LRU 是一種 Cache 替換策略，它根據 cache 中最近使用的項目，保留最新使用的內容，而淘汰最

不常使用的內容。在 SSD Cache 中，LRU 可用於管理緩存中的數據頁面，以確保最常用的數據保持在 Cache 中。

四、架構流程



五、工具使用說明

(一) Tensorflow

TensorFlow (深度學習框架) 是一個開源機器學習框架，由 Google Brain 團隊開發，用於構建和訓練深度學習模型。它提供了豐富的工具和資料庫，用於實現深度 Q-Learning 和深度神經網路。

(二) Python

Python 是一個高級程式語言，廣泛用於數據科學和機器學習領域。在這個專題中，Python 被用來實現 Deep Q-Learning、LRU 緩存管理，以及執行實驗和分析。Python 提供了簡潔的語法和廣泛的資料庫，使其成為進行實驗和研究的理想工具。

六、結論

本研究旨在使用深度強化學習 (Deep Q-Learning, DQL) 改善 SSD Cache 中的 Garbage Collection (GC) 效能。我們使用 Python 和 TensorFlow 框架實現了 DQL，並將其與傳統的 Q-Learning (QL) 進行比較。結果顯示，目前尚未證實 DQL 能夠有效提高 GC 效能，可能需要進一步的參數調整和改進。雖然研究成果尚不明顯，但本研究為未來改進 SSD Cache 的性能提供了一個有價值的起點。這需要更多研究來進一步調查和優化，以實現更好的效果。