大型語言模型推理階段記憶體負載與吞吐量權衡特性分析：離線載入與權重壓縮之影響

Characterizing Memory Overhead and Throughput Trade-offs in LLM Inference: Impacts of Offloading and Weight Compression

指導教授：何建忠
專題成員：邱繼揚、洪英豪
開發工具：Python、PyTorch
測試環境：Linux Ubuntu 24.04

一、簡介：

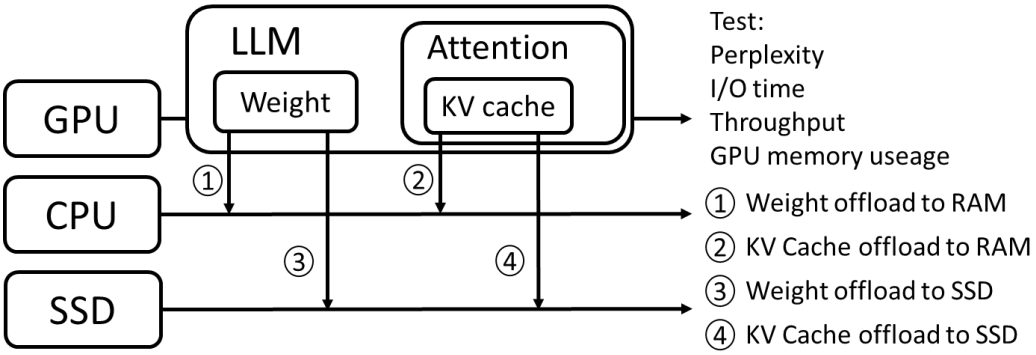
隨著大型語言模型（Large Language Models, LLMs）在自然語言處理領域的迅速發展，其龐大的參數量與記憶體需求對推理系統造成嚴峻挑戰。本研究旨在系統性分析 LLM 推理過程中記憶體負載與吞吐量（Throughput）之間的權衡特性，並探討以下兩項優化技術的效能與限制：

1.模型權重與 KV Cache 離線載入（Offloading）：透過 DeepSpeed ZeRO-Inference 與 FlexLLMGen 框架，將模型權重與 KV Cache 分別卸載至 CPU RAM 或磁碟（SSD），以降低 GPU 記憶體使用。

2.模型權重量化（Weight Quantization）：針對低位元精度格式進行實驗，觀察其對模型推理吞吐量與精度的實際影響。

我們使用 DeepSpeed 和 FlexLLMGen 這兩個支援 offloading 與記憶體最佳化的推理框架，進行一系列模型推理測試，涵蓋以下關鍵指標：困惑度、推理過程中模型各階段的消耗時間等等。

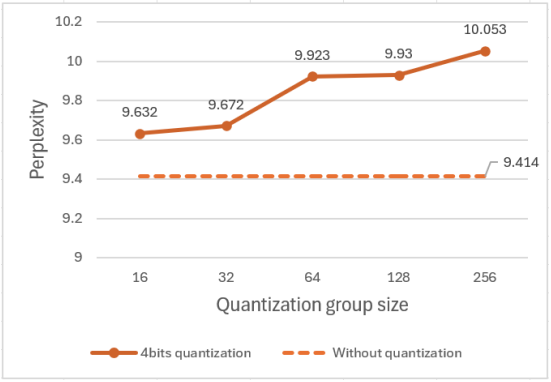
以下為系統架構圖：



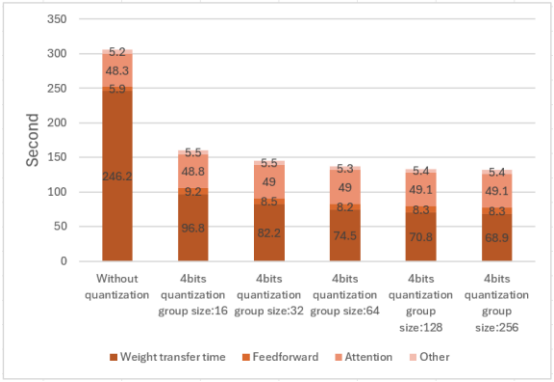
二、測試結果：

量化技術可以將模型權重以較低的位元數（如 4-bit）表示，大幅降低記憶體使用量。在進行量化時，常採用分組量化來減少精度損失，其作法是將權重劃分為小組，並為每組獨立計算量化參數，以在壓縮模型的同時維持較佳準確率，然而，分組越細，雖然可降低精度損失，卻也需儲存更多的量化參數，造成額外的儲存開銷。

我們使用 Deepspeed 對 LLaMA-2-7B 模型進行實驗，將量化後的權重儲存於 CPU RAM 中，並比較未量化模型與不同 group size 設定下的表現差異。圖一為困惑度（perplexity）的比較，圖二為推理過程中模型各階段的消耗時間比較。

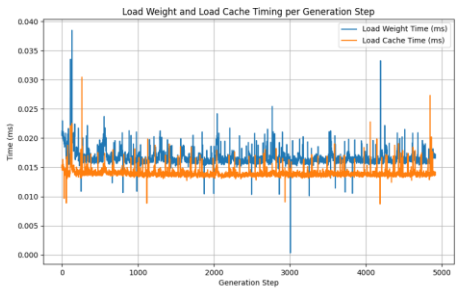


（圖一）

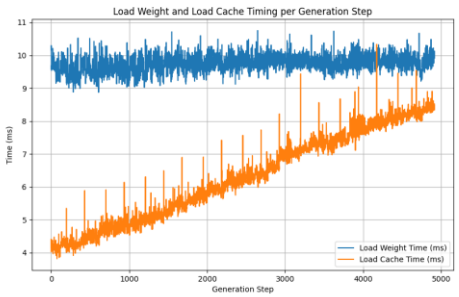


（圖二）

在 FlexLLMGen 的實驗中，比較了載入權重與 KV Cache 的時間，圖三為未啟用 offload ，模型的權重與 KV Cache 全部常駐於 GPU 記憶體中，因此每一步的載入時間幾乎可以忽略，推理速度穩定且快速。相對地，圖四為在啟用 offload 至 CPU RAM 的情境下，每步驟需從 CPU RAM 載入模型權重與 KV Cache，導致載入時間大幅上升。其中 Load Cache Time 隨步驟數線性增加，反映出 KV Cache 隨上下文長度變長而快速增長，進一步造成 I/O 負擔。



（圖三）



（圖四）