KV 快取卸載與 NetApp

NetApp artificial intelligence solutions

NetApp
August 20, 2026

目錄

KV 快取卸載與 NetApp

第一波企業人工智慧投資的重點在於模型訓練和微調，旨在建立功能，而非大規模應用。隨著企業從實驗階段過渡到生產階段，重心轉向即時推理：使用者持續互動的搜尋助理、聊天機器人、副駕駛和代理程式工作流程。在這些環境中，GPU 使用率迅速上升，並非因為每個請求都會執行完整的訓練過程，而是因為每個後續問題、對話中的每個新環節以及每個並行使用者都會增加推理堆疊的負載。

介紹

很快，一個規律就顯現出來：GPU 執行了大量重複工作，對已經處理過的標記重新計算注意力。這種重複不只是調校問題，更是記憶體容量架構的問題。產業的回應是圍繞 KV 快取所建構的分層記憶體容量策略。

什麼是 **KV** 快取？

簡而言之，KV（鍵值）快取是一種透過將先前計算的值儲存在 KV 快取中來最佳化大型語言模型 (LLM) 推論的技術，如此一來就不需要為每個新產生的 Token 重新計算這些值，否則便必須如此。

注意：如需更深入的技術總覽，請參閱 "Hugging Face KV 快取總覽"。

什麼是 **KV** 快取卸載？

大多數推理引擎預設將 KV 快取儲存在 GPU 記憶體容量中。在需求成長之前，這種方式運作良好。KV 快取大小與批次大小（同時處理的提示數量）和序列長度（每個對話或生成串流的長度）呈線性關係。隨著上下文視窗擴充、多輪對話日益普及，以及越來越多的使用者和代理程式使用推理服務，KV 快取需求可能很快就會超過可用的 GPU 記憶體容量。在這種情況下，有用的快取項目通常會被清除，引擎必須重新計算已處理詞元的注意力。

KV 快取卸載透過將先前處理過的提示的 KV 快取移至 GPU 或加速器記憶體容量之外的外部目標（例如系統 RAM、本機磁碟或共享儲存設備）來解決此限制。卸載的項目可保留至容量允許為止，並在遇到類似的前綴或後續請求時再次引用，而非在 GPU 記憶體容量填滿時遭到捨棄。實際上，這些卸載目標充當 GPU 記憶體容量的階層式交換空間：速度比 VRAM 慢，但容量更大、持久性更高，從而擴充系統在無需重複預填充工作的情況下能夠維持的對話上下文與並行工作負載。

共享儲存設備分層的重要性

當單一推理引擎的 GPU 記憶體容量不足時，KV 快取卸載就能發揮作用。將快取區塊移至 CPU RAM 可擴充單一伺服器的容量。這固然有用，但僅解決了生產環境中的部分問題。實際的推理平台很少以單一服務引擎執行個體的形式運作。它們通常運行於負載平衡器之後，進行橫向擴充，並服務於眾多並行使用者和代理程式。在這種情況下，共享儲存設備才能將 KV 快取從本機最佳化轉變為平台級功能。

- 共享儲存設備支援跨服務執行個體的重複使用 共享儲存設備的主要優勢之一是能夠跨多個執行個體和節點存取相同的檔案或物件。對於負載平衡器後方部署兩個或多個服務引擎執行個體的擴充部署而言，這一點尤其重要，因為每個執行個體都設定為將 KV 快取區塊卸載到同一個共享儲存桶或 NAS Volume。例如，當一個執行個體處理提示前綴並卸載生成的快取區塊時，另一個執行個體可以在快取命中時載入這些區塊，而無需重新計算相同的預填充工作。這一點至關重要，因為生產流量是分散式的，使用者的第一個問題可能存取執行個體 A；後續問題或其他具有類似系統提示的使用者可能存取執行個體 B。如果沒有共享儲存設備，每個執行個體都將維護自己的私有快取。

- 僅靠 **CPU** 記憶體容量不足以支援多執行個體部署 CPU 分層速度很快，但它僅限於每個服務引擎進程。除非實作點對點通道，否則一個執行個體無法存取另一個執行個體卸載到其本機 CPU RAM 中的 KV 快取條目，而點對點通路會引入額外的延遲和維運複雜性。共享儲存設備消除了這一障礙。CPU RAM 在每個節點上仍然作為快速接移分層發揮重要作用，但共享的 S3 或 NAS 提供了多個引擎可以讀取和寫入的共用記憶體容量平面。您不再是隔離擴充 GPU，而是透過共享的快取基礎架構進行擴充。
- 支援三層設計策略 + 理想的架構結合了：
  - **GPU** 記憶體容量 — 用於處理進行中請求的作用中快取。
  - **CPU** 記憶體容量 — 當提示進入佇列時快速接移。
  - 共享儲存設備 — 持久性、大容量、可共享的後備儲存設備。

  透過這種架構，當新的請求到達時，KV 快取區塊可從共享層接移至 CPU 記憶體容量中，使 GPU 專注於作用中工作，同時在儲存設備上保留持久性快取。

- 推理的經濟性，而不僅僅是速度 從生產角度來看，共享儲存設備的重要性不僅在於延遲。還在於對記憶體容量、並行性和成本的控制。像 vLLM 這樣的服務引擎可以在單一節點內有效率地管理 KV 快取。像 LMCache 這樣的 KV 快取卸載架構可以將 KV 快取轉換為可移植、可共享的資產，使其能夠在 CPU 記憶體容量和儲存設備之間移動。像 NetApp ONTAP 和 StorageGRID 這樣的共享儲存設備平台提供了持久分層，使跨多個執行個體的共享成為可能。透過將 LMCache 連接到共享儲存設備，多個 vLLM 執行個體可以存取相同的卸載 KV 快取項目，從而提高處理量並隨著並行性的增加減少備援計算。這直接減少了重複預填充造成的 GPU 週期浪費，對於聊天機器人、搜尋助理、代理程式，以及任何具有多輪執行緒、共享的系統提示或重複上下文模式的工作負載而言尤為關鍵。
- 提升推理效能和 **GPU** 投資回報 + 此基準測試比較了隨著對話長度和並行負載增加時的推理效能。當 KV 快取僅保存在 GPU 記憶體容量中時，可用的記憶體空間會迅速耗盡，處理量也會下降（橘色線）。透過三層設計（GPU 用於作用中工作、CPU 用於快速接移、共享儲存設備用於持久且可共享的快取），該平台可支援更多工作階段，並避免相同程度的效能急劇下降。實際上，分層透過減少重複的預填運算，幫助您從相同的 GPU 獲得更多工作負載。
  - 簡單來說：
    - **GPU tier** — 處理作用中的進行中工作。
    - **CPU tier** — 將最近使用的快取保存在服務引擎附近。
    - 共享儲存層 — 跨伺服器保留快取，並釋放 GPU/CPU 記憶體容量以供新工作階段使用。

  圖 1 - 多輪推理基準測試

基準測試表明，在 CPU 記憶體容量之外新增共享儲存設備並不會降低效能；相反，它擴展了處理量下降之前的可用操作範圍。分層有效地為推理新增記憶體容量階層，從而減少了每次工作階段數、上下文長度或並行使用者數增加時都需要新增 GPU 的需求。

- 企業級適用性：標準協定，無需專屬用戶端 + 共享儲存設備固然重要，但並非所有共享儲存設備都一樣。NetApp 支援使用業界標準協定和工具進行 KV 快取卸載：
  - **StorageGRID S3**
  - 適用於 **ONTAP NAS** 的 **NFS / pNFS**

  無需自訂專屬推理用戶端。維運團隊可以使用與其他資料服務相同的儲存設備協議，並享有熟悉的資料保護、安全性和多雲移動性。

部署 KV 快取卸載

共享儲存設備可擴充 KV 快取容量，並支援跨服務執行個體的重複使用。若要在正式作業環境中使用該分層，您需設定推理引擎和 KV 快取卸載架構。以下章節將說明如何使用常用工具選項來實作此堆疊。

vLLM

vLLM 是一款受歡迎的高效能開源 LLM 推理引擎。在本方案中，它負責接收用戶端請求、產生回應，並在推理過程中管理 GPU 記憶體容量中的 KV 快取。vLLM 具有可插拔性—您可以選擇使用哪種卸載架構。以下章節將介紹如何使用常見的卸載架構來實作基於 vLLM 的服務堆疊。

LMCache（進程內模式）

LMCache 是一個廣受歡迎的開源 KV 快取卸載架構。本節說明如何實作以 vLLM 為基礎的服務堆疊，該堆疊使用 LMCache 進行 KV 快取卸載。在此實作中，LMCache 與 vLLM 協同運作，將 KV 快取從 GPU 記憶體容量移至更大的儲存層，例如 CPU RAM、本機磁碟或共享儲存設備（例如 StorageGRID S3 或 ONTAP NAS 裝載）。

在預設設定下，vLLM 僅將 KV 快取保存在 GPU 記憶體容量中。隨著系統負載增加，這會成為瓶頸。本方案將 vLLM 與 LMCache 結合使用：vLLM 負責處理模型，而 LMCache 則負責將快取卸載並重複用於 CPU 和共享的 NetApp 儲存設備。LMCache 透過連接器與 vLLM 連線；啟動時，您可以使用 vLLM 的 --kv-transfer-config 標誌啟用該連接器，以便 vLLM 和 LMCache 將快取儲存到儲存設備中，並在快取命中時重新載入。

進程內模式是使用 vLLM 運行 LMCache 的原始方法。使用進程內模式時，LMCache 會在 vLLM 進程內運作。後續版本的 LMCache 新增了多進程模式，目前建議使用多進程模式以獲得更好的功能支援和效能。本節介紹進程內模式，該模式在目前的 LMCache 文件中已被標記為已棄用。對於新的部署，請考慮改用多進程模式。

本次部署，您將：

- 將 vLLM 與 LMCache 一起安裝
- 啟動 vLLM 時啟用 LMCache 連接器
- 部署三層設定（GPU + CPU + 共享儲存設備），其中兩個 vLLM 執行個體（位於分隔的 GPU 上）位於 vLLM 路由器之後，兩者均使用相同的共享儲存設備後端。

先決條件

- Linux GPU 伺服器，配備 NVIDIA 驅動程式（包括 CUDA）和至少一個 GPU（完整的雙執行個體 + 路由器佈局需要兩個 GPU 或多個伺服器）
- Python 和 uv 已安裝
- 已安裝 Docker 和 NVIDIA Container Toolkit（步驟 4 中的 vLLM 路由器需要）
- 需要網路存取才能下載模型（例如，Hugging Face）並存取您的儲存設備端點

StorageGRID S3 後端

- 建立用於 KV 快取的 S3 儲存桶
- 儲存桶的讀取/寫入憑證
- GPU 伺服器到 S3 端點的網路連線
- 已啟用虛擬託管式 S3 請求

ONTAP pNFS/NFS 後端

- ONTAP Volume 已匯出至 GPU 伺服器
- 在執行 vLLM 的*每台*伺服器上建立裝載路徑（例如 /mnt/kvcache）。高效能 pNFS over RDMA (RoCE) 的裝載命令範例：

```
mount -o
vers=4.1,proto=rdma,trunkdiscovery,write=eager,rsize=262144,wsize=262144
my_storage.my_company.net:/kvcache /mnt/kvcache
```

[[1-install-vllm-and-lmcache]]
=== 1.安裝 vLLM 和 LMCache

建立虛擬化環境並安裝 vLLM 和 LMCache。如需詳細資訊，請參閱 LMCache "安裝文件"。請務必依照與您的 CUDA 版本相符的正確安裝路徑進行安裝。

```
uv venv .venv
source .venv/bin/activate
uv pip install --upgrade lmcache vllm
```

至此，您將擁有推理引擎（vLLM）和快取層（LMCache）。

[[2-configure-lmcache]]
=== 2.設定 LMCache

vLLM 本身不會卸載 KV 快取。您需要透過 vLLM 的 KV 傳輸連接器啟用 LMCache。建立一個 YAML 檔案（lmcache-config.yaml），用於定義 CPU 和共享儲存設備分層的組態。LMCache 會在 vLLM 推理啟動序列中讀取此 YAML 檔案。

範例程式碼片段：**StorageGRID S3** 共享的分層

```
local_cpu: True
max_local_cpu_size: 100
save_decode_cache: True
remote_url: "s3://<bucket>.<storagegrid_s3_endpoint_fqdn>"
remote_serde: "naive"
extra_config:
  s3_num_io_threads: 320
  s3_prefer_http2: False
  s3_enable_s3express: False
  save_chunk_meta: False
  disable_tls: <True/False>
  aws_access_key_id: "<storagegrid_s3_access_key>"
  aws_secret_access_key: "<storagegrid_s3_secret_key>"
  s3_region: "us-east-1"
```

注意：請將儲存桶名稱、端點、憑證和 `disable_tls`替換為符合您環境的內容。

範例片段：**ONTAP pNFS/NAS** 共享的分層

```
local_cpu: true
max_local_cpu_size: 100
save_decode_cache: true
remote_storage_plugins: ["fs"]
extra_config:
  remote_storage_plugin.fs.base_path: /mnt/kvcache
  save_chunk_meta: false
```

在步驟 3 之前，使用您站台的 NFS/pNFS 程序裝載 ONTAP 匯出。

[[3-run-two-vllm-instances-shared-cache-across-servers]]
=== 3.執行兩個 vLLM 執行個體（伺服器間共享的快取）

在兩個執行個體上設定通用環境變數：

```
export LMCACHE_CONFIG_FILE=$(pwd)/lmcache-config.yaml
export PYTHONHASHSEED=0
export VLLM_API_KEY='<shared_api_key>'
```

執行個體 **1**（GPU 0，連接埠 8001）：

```
export CUDA_VISIBLE_DEVICES=0
vllm serve Qwen/Qwen3-8B \
  --tensor-parallel-size 1 \
  --kv-transfer-config
'{"kv_connector":"LMCacheConnectorV1","kv_role":"kv_both"}' \
  --port 8001
```

Instance 2（GPU 1，連接埠 8002 — 分隔終端）：

```
export CUDA_VISIBLE_DEVICES=1
vllm serve Qwen/Qwen3-8B \
  --tensor-parallel-size 1 \
  --kv-transfer-config
'{"kv_connector":"LMCacheConnectorV1","kv_role":"kv_both"}' \
  --port 8002
```

使用相同的設定檔案和雜湊種子，以便兩個執行個體對快取區塊識別達成一致，並能從共享儲存設備中讀取彼此卸載的資料。在兩個執行個體上設定 `VLLM_API_KEY`；路由器會將此值以 `OPENAI_API_KEY`的形式傳遞，以便路由請求能夠被執行個體接受。

注意：單一 GPU 執行個體也可用於實作階層式儲存架構。在這種執行個體中，請跳過步驟 4。

[[4-deploy-the-vllm-router-single-entry-point]]
=== 4.部署 vLLM 路由器（單一入口點）

在兩個執行個體都運作正常後，部署一個路由器，以便用戶端使用一個與 OpenAI 相容的 URL。

```
docker run -d --rm --name vllm-router \
  --network host \
  -e OPENAI_API_KEY="${VLLM_API_KEY}" \
  ghcr.io/vllm-project/production-stack/router:latest \
  --port 8000 \
  --service-discovery static \
  --static-backends "http://localhost:8001,http://localhost:8002" \
  --static-models "Qwen/Qwen3-8B,Qwen/Qwen3-8B" \
  --routing-logic roundrobin
```

將流量傳送到 http://localhost:8000/v1。路由器分發請求；LMCache 和共享儲存設備允許跨執行個體重複使用快取，而不僅限於單一 GPU 內。

例如，您可以使用 curl 向路由器發送請求，如下所示（請將 <shared_api_key> 替換為您對應的 API 金鑰）：

```
curl -s http://localhost:8000/v1/chat/completions \
  -H "Content-Type: application/json" \
  -H 'Authorization: Bearer <shared_api_key>' \
  -d '{
    "model": "Qwen/Qwen3-8B",
    "messages": [{"role": "user", "content": "What is 2+2?"}]
  }' | jq .
```

[[5-validate-that-tiering-is-working]]
=== 5.驗證分層是否有效

- vLLM 的兩個進程都傾聽 8001 和 8002 連接埠；路由器在 8000 連接埠上回應。
- 透過路由器傳送含有長前綴的請求。
- 確認物件或檔案出現在共享儲存設備（S3 儲存桶或 ls -ltr /mnt/kvcache）上。
- 再次發送類似的請求——第二次請求應在日誌中顯示更快的預先填充或快取命中率行為。
- 透過路由器重複此操作，以便流量可以到達另一個執行個體。

結論

部署階層式鍵值快取架構可將鍵值快取從 GPU 記憶體容量轉移到 CPU 記憶體容量和共享的 NetApp 儲存設備，使推理能夠隨使用者數量和上下文長度擴充，而無需浪費 GPU 週期進行重複預填。共享儲存設備將快取轉化為跨服務執行個體的可重複使用基礎架構，並協助您從現有的 GPU 投資中獲得更多價值。

NetApp 儲存設備非常適合此分層，因為它除了提供原始容量之外，還能滿足生產推理所需的其他功能：透過 S3 和 NFS/pNFS 進行標準存取、多個服務引擎執行個體可以同時使用的共享的快取，以及您已依賴的企業資料

服務，確保持久性、保護和治理功能。您無需使用專屬用戶端；KV 快取卸載架構使用熟悉的協定連接至 StorageGRID S3 或 ONTAP NAS 裝載。

NetApp 為您提供熟悉的儲存基礎，用於 KV 快取卸載，而無需改變您執行推理的方式或您的團隊管理資料的方式。

版權資訊

商標資訊