jh251003

機械学習向けストレージアーキテクチャの研究

中村隆喜（東北大学 サイバーサイエンスセンター）

概要

機械学習を用いたデータ処理では、GPU メモリ、メインメモリが有限サイズであることなどから、ストレージの活用が必須である。学習データセットや学習モデルの大規模化に伴って、ストレージの活用がより重要となる。したがって、これらの処理を考慮したストレージアーキテクチャを確立する必要がある。本研究では、機械学習を用いたデータ処理におけるストレージ観点での課題を明らかにし、その解決方法についての検討を行う。

機械学習のひとつであり、近年急速に利用が進んでいる大規模言語モデルは、その知識を拡大するために Retrieval Augmented Generation (RAG)、及びファインチューニングと呼ばれる手法が用いられるのが一般的である。新規課題にあたる今年度は、これらの手法に着目し、ストレージ観点での課題に取り組んだ。その結果、RAG 向け精度混合型ベクトルデータベース(Chimera-VDB)の発明、ファインチューニング時のメモリ使用量低減手法の確立など、多くの成果を創出した。

1. 共同研究に関する情報

(1) 共同利用・共同研究を実施している拠点名

東北大学 サイバーサイエンスセンター

mdx I

(2) 課題分野

データ科学・データ利活用課題分野

(3) 参加研究者一覧と役割分担

中村隆喜(東北大学)：統括、ストレージ観点の分析。具体的には、LLM-RAG 向けの精度混在ベクトルデータベースの開発、機械学習時の必要メモリ量を低減するストレージオフロード機能の検討、機械学習用データを格納するオブジェクトストレージの性能安定化等。

亀井仁志(香川大学)：システムソフトウェア観点の分析。具体的には、AI 基盤向け新データ保護方式の研究開発。

安藤一秋(香川大学)；機械学習観点の分析。

また、増田嶺、岩本和真、山根直、矢野揮一、Xiaojie Tan が開発や検証を実施した。

2. 研究の目的と意義

機械学習を用いたデータ処理では、GPU メモリや計算機のメインメモリ上のデータにアクセスし、操作することで高速な処理が実現されている。しかしこのメモリ容量は有限であり、データの揮発性があるという特徴から、補助記憶装置であるストレージの活用が必須である。

機械学習の一部である大規模言語モデルの処理においても、ストレージの活用が必要なケースはいくつかあげられる。図１にその代表例を示す。まず、モデルの学習時には、学習用データセットの読み込み、学習中モデルのチェックポインティング、学習済みモデルの保存がある。また、Retrieval Augmented Generation(RAG)を用いた推論時には、RAG インデックス情報のデータベース検索、クエリに適合する RAG 参照情報の読み込み、学習済みモデルの読み込みがある。

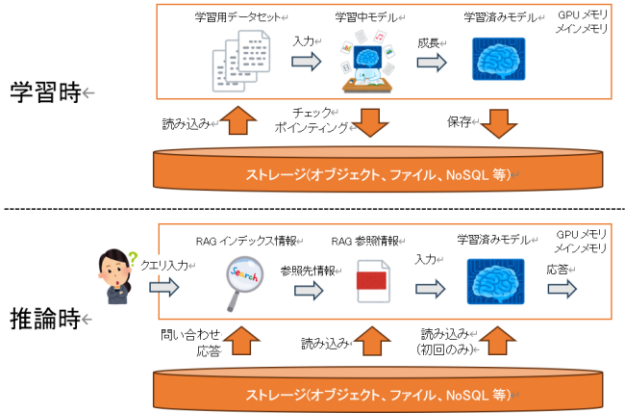

図 1　機械学習におけるストレージ活用代表例

これ以外にも、RAG データの追加時には、データベースへの RAG インデックス情報の追加と RAG 参照情報の保存がある。

学習データセットや学習モデルが大規模化するに伴って、ストレージの活用がより重要になる。したがって、これらの処理を考慮したストレージアーキテクチャを確立する必要がある。本研究では、機械学習を用いたデータ処理におけるストレージ観点での性能やデータ保護上の課題を明らかにすること、またそれら課題を解決する方法についての検討を行った。本研究はシステムの基盤部分に焦点を当てていることから、その成果は IT 基盤技術としての活用及びデータ利活用分野への幅広い応用が期待できる。

本共同研究における学術的意義は、ストレージ、システムソフトウェア観点での性能分析と大規模言語モデル処理の深い理解に基づき進める必要があり、情報分野内で極めて学際的な取り組みである点にある。

本共同研究における社会的意義は、現在急速に利用が進んでいる RAG システムを対象としており、実社会システムに技術がそのまま応用可能な点にある。

3.　当拠点の公募型共同研究として実施した意義

データ活用社会創成プラットフォーム協働事業体が運用しているデータ活用社会創成プラットフォーム mdx I は、本課題を実行するために最も適している環境である。具体的には、NVIDIA Tesla A100 を持つ GPU 演算加速ノードを多数保有し、汎用 CPU ノードを多数保有し、VMware ベースの仮想化環境により OS を占有環境にでき、仮想ディスク・高速内部ストレージ・大容量内部ストレージ・オブジェクトストレージといった複数種類のストレージを保有しているという点で適している。

本研究では、RAG 向けの埋め込みベクトル生成処理、Fine-Tuning システムの構築、推論システムの構築等でオープンソースのローカル機械学習モデルの利用を予定しているため、先進的な GPU を持った GPU 演算加速ノードが必要となる。また、機械学習向けのストレージシステムを構築するためには、複数の種類のストレージとそのコントロール機能となる汎用 CPU ノードが必要となる。OS レイヤでの分析や改造を行うためには、OS 自体が入れ替え可能な IaaS 形式の仮想化環境が必要となる。これらの環境は商用クラウドでも利用可能であるが、同等の実験を行うためには多額な費用が必要となるため、本共同研究の枠組みを活用して実施した。

4.　前年度までに得られた研究成果の概要

新規課題のため該当なし

5.　今年度の研究成果の詳細

今年度の代表的な研究成果について、以下 5 点述べる。また、最後に計算機利用上の工夫について述べる。

(1) 精度混在型ベクトルデータベース Chimera-VDB の提案と開発

LLM-RAG システムの拡大に伴い、参照情報の特徴量ベクトルを格納するストレージ容量の増加が懸念されている。これに対し、インデックス構造に由来する検索上の重要度に応じて特徴量ベクトルの表現精度を決定し、その混在環境で検索を行う方式（Chimera-VDB）を提案している。提案のアーキテクチャを図 2 に示す。

表現精度の決定方法として、インデックスの

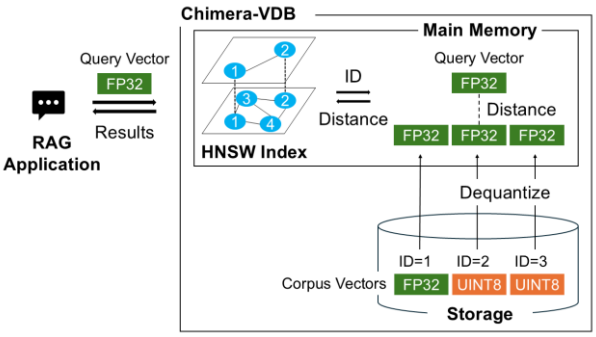

図 2 Chimera-VDB のアーキテクチャ

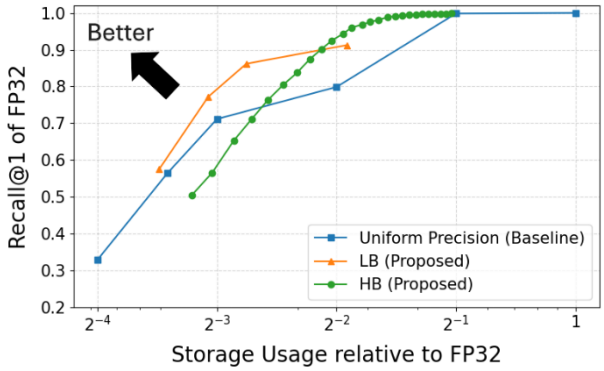

図 3 提案方式による検索再現度向上効果

階層構造に着目した LB アルゴリズムと、ノード間の接続関係に着目した HB アルゴリズムの 2 種類を提案している。結果は図 3 の通りであり、提案方式の LB、HB がともに、全てのベクトルの表現精度を同じビット数に揃える既存方式に比べ、より少ないストレージ消費でより高い検索再現度を達成している。本成果は国際ワークショップ 16th ACM SIGOPS Asia-Pacific Workshop on Systems で発表した(図 4)。

また、提案方式の時間観点での評価を行った。具体的には精度混在型ベクトルデータベースの構築時間及び検索時間を処理段階ごとに詳細に評価した。本成果は 2026 年 2 月開催の情報処理学会 OS 研究会で発表した。

さらに、量子化を含む精度混在を行った場合に発生する「類似度シフト」と呼ばれる問題を分析し、検索再現度をさらに向上させる方式の検討を行った。この検討結果をストレージ分野の産学コミュニティメンバが集うトップカンファレンスである 24th USENIX Conference on File and Storage Technologies (FAST 2026)の Work-in-Progress セッションで発表した(図 5)。

図 4 APSys2025(@韓国ソウル)発表の様子

図 5 FAST 2026(@USA, CA)発表の様子

これに加えて、複数のデータセットを用いた場合の提案方式の効果の評価も実施し、現在 FAST 2027 の Spring Deadline に投稿中である。

(2) ファインチューニング時におけるコールドデータオフローディング機能の提案と開発

大規模言語モデル(LLM)のファインチューニング時には膨大な GPU メモリが必要となる。その大部分をオプティマイザステートと呼ばれる変数が占めているものの、フォワードパスやバックワードパス処理中はアクセスされない。メモリ利用効率を向上させるため、我々の研究グループは NVIDIA GPUDirect Storage を用いて、これらのコールドな状態を NVMe ストレージにオフロードする手法を確立した。最適化状態を CPU 経由で NVMe にオフロードし、CPU 上で最適化の

図 6 第 101 回情報システム研究会発表の様子

図 7 第 103 回情報システム研究会発表の様子

更新を行う DeepSpeed ZeRO-3 とは異なり、本手法では、更新処理を GPU 上で維持しつつ、GPU と NVMe の間で最適化状態を直接転送することで CPU をバイパスする。最適化状態は各更新ステップ後に同期的に NVMe にオフロードされ、次の更新前にリロードされるため、フォワードパスおよびバックワードパス処理中にメモリに常駐することはない。BERT-Large を用いたファインチューニングの測定により、ZeRO-3 の NVMe オフロード方式と比較して、提案手法は、トレーニング処理スループットを 29.2%向上させ、CPU メインメモリ使用量を 85.2%削減することを実証した。さらに、提案手法は、標準的な非オフロードでの GPU トレーニングと同等の収束性を維持しつつ、GPU メモリのピーク使用量を 28.0%低減することに成功した。

これらの成果のうちファインチューニング時におけるメモリの時系列分析結果を、電気学会第 101 回情報システム研究会で発表した(図 6)。また、ファインチューニング処理スループットとメモリ削減の評価結果については、情報処理学会 2026 年 5 月 OS 研究会にて発表予定である。

(3) オブジェクトストレージを活用した分散ファイルシステムの初期検討

機械学習の発展に対し、個々の研究者や組織が利用する計算機資源は不足状態にある。このような状況において、限られた資源を最大限に活用するため、複数の計算機やストレージを組み合わせた運用の形態が広がりを見せている。具体的には、ある GPU 計算機で機械学習を途中まで行い、中間状態のデータをストレージに保存し、残りの学習を別の GPU 計算機で行うといった、フェーズや目的に応じて渡り歩くワークフローである。また、スーパーコンピュータで物理化学シミュレーションを行い、得られたシミュレーション結果を機械学習の学習データとして別の GPU 計算機で用いるワークフローもある。このように複数の計算機をまたいで学習を行う場合、GPU や CPU の性能を最大限に発揮するために I/O 待ちの状態を避けて、遠隔地を含む複数計算機間で数テラバイト規模の学習データを短時間で共有する必要がある。さらに複数の計算機間でデータの不整合を防ぐためにデータの一貫性を保つことも必要である。

そこで我々の研究グループでは、複数の計算機から高速かつデータの一貫性を保ったデータの共有を行う分散ファイルシステムの実現を目的として研究を進めている。この目的を達成するためにオブジェクトストレージのマウントツールである s3fs を用いて、その特性を分析することによって I/O 性能の向上についての検討を行った。分析した結果に基づき、オブジェクトストレージへのアップロード処理を非同期処理に改良したところ、1GB のファイルの書き込み時のファイルの open()から close()までの時間を従

来の 6 秒程度から 2.31 秒短縮することに成功した。

これらの成果は、電気学会第 103 回情報システム研究会で発表した(図 7)。

(4) RAG 検索高速化キャッシュ（TC-HNSW）

近年、生成 AI を用いた検索として、Retrieval Augmented Generation(RAG)が注目されている。RAG は企業内部の情報などを利用して、LLM による検索より高度な検索を実現する。

RAG のシステムは、LLM だけでなくベクトル DB など、複数のサブシステムを組み合わせて構成される。そのため、検索結果の応答時間が長い問題があった。

検索結果応答時間を分析した結果、ベクトル DB の処理時間が問題であることが分かった。そこで、主要なベクトル DB の OSS である Qdrant を対象とし、「トピックバイアス仮説」に基づくキャッシュ機能である Tired Cache HNSW(TC-HNSW)を提案して実装・評価した。トピックバイアスとは、RAG 検索におけるトピックの偏りのことであり、例えば「特定の部署では特定の話題が検索されやすい」などが該当する。

TC-HNSW を Qdrant に実装して評価した結果、検索精度を落とすことなく、特定のトピックに対して高速に応答できることを確認した。

(5) 新データ保護方式（FEC）

機械学習の対象データの保護が重要となる。我々の研究グループは、従来のデータ保護方式では全データを失う深刻な障害時にも、部分的にデータを保護できる度合い(耐久性)を高めることができる File-level Erasure Coding(FEC)と呼ぶ新しいデータ保護方式を提案している。

現在、大量のデータを保存に適したオブジェクトストレージシステムの Open Source Software(OSS) である MinIO への実装を進めている。MinIO において FEC 方式を適用するためには、オブジェクトの PUT 処理の延長で類似サイズのファイルとのペアリングを行い、そのファイルペアを耐久性の高い配置方法でストレージに書き込む必要がある。さらに、処理を逐次的に処理する方式と一時バッファに格納して処理する方式を実装した。FEC を実装した MinIO における PUT 処理の実装概要を図 8 に示す。

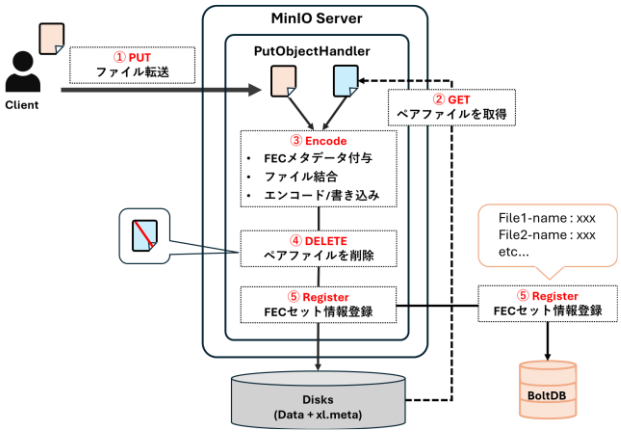

図 8 MinIO PUT 処理実装

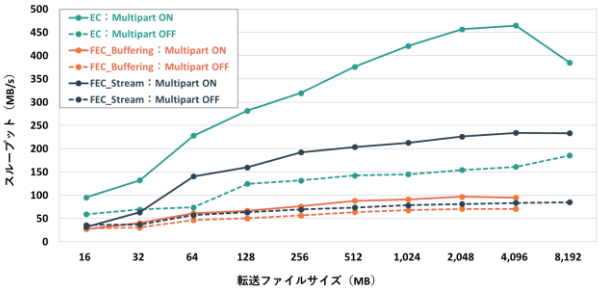

図 9 MinIO PUT 性能評価結果

FEC を実装した MinIO を用いて、2 ファイルを FEC により処理する場合における、PUT 性能評価結果を図 9 に示す。ファイルサイズを 16MB から 8192MB まで 2 のべき乗で増加させてスループットを評価した。凡例の EC は従来方式の結果である。FEC の一時バッファ方式（Buffering）は、バッファ溢れにより、8GB のデータ PUT に失敗した。FEC の逐次処理方式（Stream）は、ペアファイルを考慮すると、およそ半分の性能となった。一方、FEC のオーバヘッドを考慮すると、ほぼ最大値性能を達成した。

計算機利用上の工夫

利用ポイントやリソースの節約に努め、管理を容易にするための運用を徹底した。

具体的に節約の観点では、課題代表者が VM の

稼働状況を定期的に監視し、利用せず稼働し続けている VM の利用者に停止を促した。また、グローバル IP アドレス、仮想ディスクなど量の制約が厳しいリソースについては、定期的な棚卸しを行い、プロジェクト内での適切なリソース配分を維持した。

次に管理容易化の観点では、VM の担当者が不明にならないように、VM 名に作成者の名前を付けることを徹底した。グローバル IP アドレスについては、利用者の名前を付けることが mdx I ポータル上ではできないため、別途管理台帳を用意し、使用者はその管理台帳へ記載することを徹底した。

これらにより、mdx I において効率的なリソース利用と、プロジェクト内のメンバー間で混乱の少ない運用を実現した。

6. 進捗状況の自己評価と今後の展望

申請書で設定した今年度の計画では「RAG を用いた LLM 推論システムに関する性能分析と性能課題の解決を狙う」としていた。

本観点で自己評価した場合、(1)(4)の成果がこの狙いに直接対応する。(1)(4)については、査読付き国際会議で 3 件、国内会議で 1 件の発表を行っており、新規課題であるにも関わらず十分な成果が得られている。特に(1)については、ストレージ分野でのトップカンファレンスである USENIX FAST 2026 の Work-in-Progress セッションで発表していること、さらに同内容を拡張して FAST 2027 に投稿中であるなど、継続した挑戦を続けている。また、研究業績のその他に記載の通り、(1)の内容が評価され、(1)の研究成果を活用した共同研究を 4 月より民間企業と開始することとなった。

これに加えて、(2)(3)(5)などの次年度以降に繋がる大きな成果も達成しており、国内会議で合計 5 件の発表を実施している。

以上を総合して自己評価すると、当初の計画に対して十分な成果を上げていると共に、今後の計画に資する十分な研究の仕込みも進んでいる状況であり、計画を大きく上回った成果を達成できていると考える。

今後の展望について、(1)-(5)それぞれについて以下述べる。

(1) 精度混在型ベクトルデータベース(Chimera-VDB)の今後の展望

今後の展望としては主に評価観点の拡充と新たな手法の検討の 2 つの方向性がある。

これまで Chimera-VDB では検索再現度とストレージ使用量のトレードオフを改善することを目的として研究を進めてきた。精度を混在させる各スキームの長所と短所をさらに多面的に評価するためには、最終的な RAG 応答の正確度なども評価する必要がある。

新たな手法の方向性としては、スカラー量子化とは別のベクトル圧縮手法の適用、HNSW とは別の近似最近傍探索手法への適用が考えられる。

(2) コールドデータオフローディング機能の今後の展望

現在の提案手法の設計は、クリティカルパスにおける同期 I/O のレイテンシや、評価が単一のモデル規模に限定されていることなど、いくつかの制約がある。

今後の展望としては、転送レイテンシを低減するための非同期 I/O を用いたデータ転送処理と演算処理のオーバーラップ、より大規模なモデルやマルチ GPU 環境での評価などが考えられる。

(3) オブジェクトストレージ活用分散ファイルシステムの今後の展望

今後の展望としては、I/O 性能の多角的な検証がある。具体的には同時書き込み数を増やしたときの性能検証、複数ファイルの逐次書き込みでの性能検証などがある。また、非同期化することに伴なって、計算機間でのコンシステンシの保証方法の設計と実装が必要となる。これらを行ったうえで、従来方式との性能比較が考えられる。

(4) RAG 検索高速化キャッシュの今後の展望

今回検討した TC-HNSW は、トピックバイアス仮説に基づいたキャッシュ制御を行う。今回の評価においては、いくつかのデータにおいて、トピックバイアス仮説が有効に機能しないデータ分布が観測された。

今後の展望としては、トピックバイアス仮説が有効になる程度のクラスタリングを行い、キャッシュヒット率を向上させることがある。また、今回、実装と評価では、パラメータを固定した。一方、RAG 検索を想定する場合、パラメータは動的に変化する。そのため、固定値ではなく、トピックの変化などを捉えて、動的に制御することが考えられる。

(5) 新データ保護方式（FEC）の今後の展望

FEC は、性能を考慮した実装においては、現時点で実用レベルに達したと考えられる。今後の展望としては、以下の 2 つを考えている。

1 つ目は、MinIO の CRUD(Create, Read, Update, Delete) 操作対応である。現在は、Create 処理（アップロード処理）と Read 処理（ダウンロード処理）の対応が完了している。今後は、Update 処理、Delete 処理に対応していく。いずれの処理も、オリジナルファイルが FEC の処理により変更された後の操作となるため、技術的な課題も多い。

2 つ目は、FEC に対応した MinIO の機械学習環境への実装である。現在は、MinIO のみでの動作検証を実施している。一方、将来的に機械学習基盤へ統合することで、ビッグデータの保持や機械学習処理への利用が可能となる。

以上