公開

（事後評価・研究者用）

研　究　報　告　書

「次世代メモリデバイスによるアプリケーションの自動高速化」

研究期間： 2020 年 4 月～2022 年 3 月
研究者番号： 50241
研 究 者： 穐山　空道

1．　研究のねらい

計算機のメモリはプロセッサに比べ非常に遅いという問題がある。これは後者の性能の伸びに対して前者の性能伸びが著しく遅いからである。メモリを構成する DRAM のランダムアクセス遅延（キャッシュに乗っていないデータを要求してからそのデータが届くまでの時間）は、2007 年と 2020 年のそれぞれ時点の仕様でほぼ同一である。一方同一の期間でプロセッサの性能はシングルコア性能だけでも数倍、コア数は 32 から 64 倍になっており、合計の数値計算性能は数百倍になっている。従って計算機上で実行されるソフトウェアの性能改善のためメモリの性能改善が急務である。

Approximate Memory とは DRAM のランダムアクセス遅延を改善する新技術であり、データにビット化け（エラー）が混入することを許す代わりに従来にない性能向上が可能である。これ DRAM 内部の電気的動作に定められたタイミング制約を意図的に逸脱し、電気的に完全に安定になるのを待たずに操作する技術である。

Approximate Memory はデバイスレベルでは様々な特性などが明らかにされているものの、Approximate Memory を使いアプリケーションを実際に高速化するためには課題が残されている。そこで本研究のねらいは、Approximate Memory によってアプリケーションを自動的に高速化することである。ここで自動的とは、ユーザはアプリケーションの最終出力に許容できる誤差（例：正しい結果から 1%の誤差を許す）を指定するだけでそれを保証するために必要な制御は全てシステムがユーザの関与なしに行うことである。

2．　研究成果

（1）概要

研究のねらいを達成するため、本研究では（A）エラー制御粒度とエラー耐性粒度の違いによる Approximate Memory 実用化困難性の検証 および（B）プログラム出力の誤差を保証する制御パラメータ決定手法の研究 に取り組んだ。

研究テーマ（A）は、ACT-I 本期間で発見した Approximate Memory 実用化における課題について、これまで定性評価に留まっていたものを定量的に評価したものである。これにより当該課題が Approximate Memory 実用化の上で無視できない重要課題であることが示された。本成果はこれまで看過されていた問題を定性・定量の両面で明らかにしたものであり、課題の解決には至っていないもののその指摘・定量評価自体が大きな成果である。

研究テーマ（B）は、Approximate Memory の自動的な制御のためにプログラム出力の許容誤差からデバイスの制御パラメータを逆算する手法の必要要素である。この逆算には三段階の考慮すべきものがある（詳しくは「（2）詳細」を参照）が、そのうちこれまで最も議論が

されていない「プログラムの入力誤差と出力の誤差の関係」を理解する研究を行った。具体的には、プログラムの入力誤差と出力誤差の関係をプログラムの微分値を使って数理的に理解することを提案し、また Approximate Memory 制御の観点から微分値を求める際の課題を明らかにした。さらに課題の解決策としてサンプルデータに関して求めた微分値の再利用を提案し、どのようなアプリケーションならば再利用が可能なのかを実際のデータを用い議論した。コンピュータアーキテクチャ研究では実際の現象に基づいて帰納的に手法を提案する方法論が多い中、本研究は入出力の誤差の関係の理解に数理的な手法を持ち込んだ点で価値が高い。

（2）詳細

研究テーマ A「エラー制御粒度とエラー耐性粒度の違いによる Approximate Memory 実用化困難性の検証」

本テーマでは、ACT-I 本期間で発見・指摘したアプリケーションのデータが持つエラー耐性の粒度と Approximate Memory においてエラー率を制御できる粒度が異なることに起因する問題をさらに深ぼりし、本問題が実際に重要な問題であることを示した。

アプリケーションのデータが持つエラー耐性の粒度は、数バイトの単位になりえる。例えばグラフのノードを構造体で表現し、ノード同士をポインタで繋いでグラフを表現する例を考える。このケースでは 1 ビットのエラーも許されないポインタというデータと、よりエラー耐性の高いデータ（例えばノードの評価値）が 1 つの構造体に格納されメモリ上近接する。

しかし Approximate Memory ではエラー率を制御できる粒度はより大きく、512 バイトや数キロバイトである。これは低速なメモリが高速なプロセッサに追いつくために多くのビットを同時に駆動する必要性からくる制約であり、回避することは極めて難しい。

ACT-I 加速期間では本問題の重要性を証明するため、以下の 2 つの貢献をした:

(1) ACT-I 本期間で行っていた現実のコード分析による類似ケース発見の強化

(2) 既存のメモリレイアウト変換技術による性能低下を評価し、Approximate Memory の利点を打ち消してしまう可能性があることを示した。

貢献 (1) では新たに SPEC CPU 2017 のメモリアクセスパターンとソースコードを分析し、SPEC CPU 2006 と同様の傾向があることを確認した。

貢献 (2) では既存のメモリレイアウト変換技術について調査し、それを再現するようなメモリ配置をシミュレータで再現することで性能低下を評価した。メモリレイアウト変換技術とは、構造体の各メンバをメモリ上離れた位置（アドレス）に配置することでキャッシュヒット率の向上を目指す手法である。本手法によりエラー耐性の異なるデータ（上記の例ではポインタとそれ以外）をメモリ上離れた位置に配置でき、異なるエラー率を適用できる。

メモリレイアウト変換技術はキャッシュヒット率の向上を目的としているものの、逆にキャッシュヒット率を低下させてしまう場合もある。例えばグラフの評価値に応じどの子ノードを辿るかを選ぶプログラムを考えると、構造体内の評価値と子を指すポインタは時間的に近くにアクセスされる。この時ポインタとその他のデータをメモリ上離れた位置に配置するとキャッシュヒット率低下により性能が低下する。

メモリレイアウト変換技術による性能低下を Approximate Memory による利点と比較するため、図 1（研究成果リスト(1) の 1 より引用）のようなシミュレータを開発した。本シミュレー

タは入力にレイアウト変換を適用するデータのアドレスと当該データ内のどのメンバをメモリ上離れた位置に配置するかの情報を取る。これらの情報をもとにシミュレータはプログラムコードからは完全に独立に内部的にメモリレイアウトを変換する。プログラム実行にかかったサイクル数をメモリレイアウト変換なしのケースと比較することで性能低下を評価する。

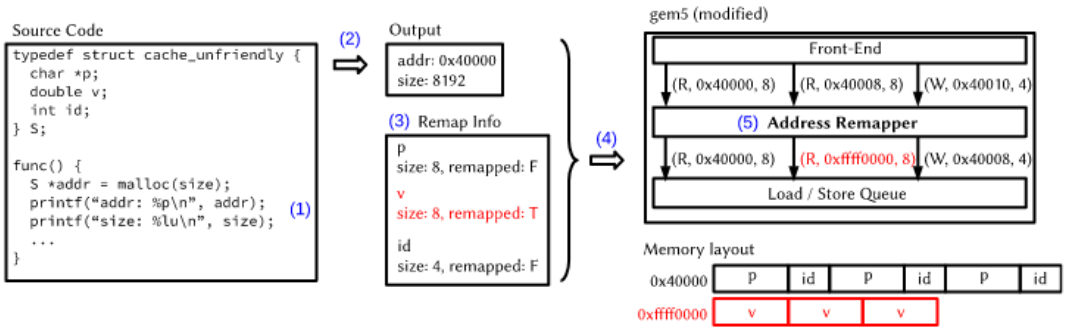

図 1：メモリレイアウト変換技術による性能低下を評価するためのシミュレータ

開発したシミュレータによる評価により、メモリレイアウト変換による性能低下が最悪ケースでは Approximate Memory の利点を打ち消すほど大きいことを発見した。

研究テーマ B「プログラム出力の誤差を保証する制御パラメータ決定手法の研究」

本テーマでは Approximate Memory 上で動作せるプログラムの出力の誤差をユーザの希望以内に抑えるための手法を探った。これを実現するためには、Approximate Memory の制御パラメータからプログラムの出力誤差への三段階の影響を逆算する必要がある。第一に制御パラメータ（タイミング制約）とビット化け発生パターンの関係、第二にビット化け発生パターンとプログラムの入力の誤差の関係、最後にプログラムの入力の誤差とプログラムの出力の誤差の関係である。

ACT-I 加速期間では三段階の影響のうちプログラムの入力の誤差とプログラムの出力の誤差の関係をプログラムの微分を用いて分析することを提案した。プログラムの微分とはプログラムを数学的な関数として微分する技術であり、入力 x のある値 xt での微分係数を計算できる。微分係数は入力 x が dx だけずれたときの出力のずれであり、プログラムの入力の誤差とプログラムの出力の誤差を表現している。

プログラムの微分を Approximate Memory 制御に利用するために、微分をどのように求めるかが課題である。例えば x^2 の微分は 2x であるように、微分は入力に依存する。従って原理的には与えられた入力に関する微分を入力値ごとに計算する必要があるが、これにはプログラムの微分の原理からプログラムの通常実行と同じ時間計算量がかかる。しかし本研究の Approximate Memory 利用目的は高速化であり、微分の計算にプログラムの通常実行と同じ時間がかかっては無意味である。

そこで文献「その他の成果」の 4 では、サンプルデータに関して求めたプログラムの微分値がどのような場合に再利用可能であるかを研究した。数学的にはプログラムが入力に対して線形である場合、非線形であっても入力にかかる比例係数がごく小さい場合には微分値

の入力への依存性が低い。しかし実際のプログラムに関してどのような場合がこれに該当するかは議論されていない。

具体的に ACT-I 加速期間では、画像処理アプリケーションを対象にプログラムの微分値を実際に算出しその変動を議論した。アプリケーションはガウシアンフィルタ、ソーベルフィルタ、離散コサイン変換、バイラテラルフィルタ、簡単なニューラルネットワークの推論である。

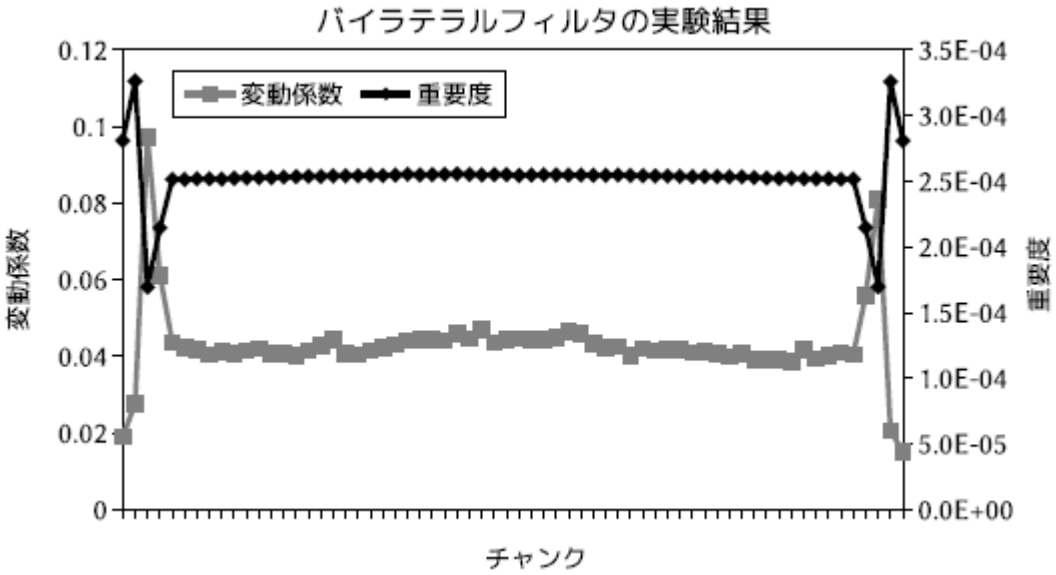

図 2:バイラテラルフィルタにおける画像のチャンク(行)の重要度とその変動係数

図 2 はバイラテラルフィルタアプリケーションについて、画像のチャンク(1 行)ごとに重要度とその変動係数を求めたものである。チャンクの重要度とは、そのチャンクに含まれる画素値がずれたときに出力値がどの程度ずれるかを示す値、変動係数とは重要度の複数の画像に対する平均を分散で割った値である。

図 2 からバイラテラルフィルタでは重要度の変動係数はおよそ 4%程度であり、本アプリケーションでは Approximate Memory 制御のためにサンプルデータに関して求めた微分値をその他のデータに関しても使いまわせる可能性が高い。詳細な実験設定やその他のアプリケーションに関する結果は文献「その他の成果」の 4 を参照のこと。

3. 今後の展開

今後の展開は (1) Approximate Memory の自動制御に向けた研究を引き続き推進すること、(2) 本研究で得られた知見をほかの Approximate Computing パラダイムにも適用し学術的波及効果を狙うこと、の二点である。

(1) について、本研究期間ではこれまで看過されていた課題を発見した点、これまで用いられてこなかった数理的手法を導入した点で独創的な成果があったものの、大きな最終目標の達成には道半ばである。 本研究期間で得た知見・技術を基に今後も目標達成に向け研究を行う。

(2) について、他の Approximate Computing パラダイムとは例えば CPU 内部の演算を高速・不正確に行うための回路研究などが該当する。例えば加算を高速・不正確に行うためには繰り上がりの伝播を途中で打ち切る手法が知られており、その制御(どこで打ち切るか)はアプリケーショ

ンの最終出力の許容誤差から逆算し決定するべきと考えている。そのためには本研究で提案したようにプログラムの入出力の誤差同士の関係を数理的な理解が有効だと予想される。

4. 自己評価

研究目的の達成状況

研究目的の達成に向け着実に進捗している。本研究の成果はいずれもこれまでほかの研究者によって議論されていなかったものであり、Approximate Memory の自動的な制御に向けて本質的な課題を世界に先駆けて明らかにしている。今後も引き続き研究を続けることで最終目標を達成できると考える。

研究の進め方（研究実施体制及び研究費執行状況）

研究費は状況応じて臨機応変に執行することができた。新型コロナウィルス感染症拡大の影響により予定していた海外滞在や国際会議での情報収集ができなくなったが、その予算を有効活用し研究を進めることができた。

研究成果の科学技術及び学術・産業・社会・文化への波及効果

研究成果の科学技術及び学術の波及効果は、Approximate Memory というハードウェア制御のためにプログラムの微分という数理的な手法を持ち込んだことが大きいと考える。コンピュータアーキテクチャ分野の多くの研究では具体的な事象に基づき改善策を提案する帰納的な方法論が多いが、本研究ではそれに一石を投じるものである。

研究課題の独創性・挑戦性

本研究課題は Approximate Memory 実用化のためにエラー発生メカニズムを考慮した上でソフトウェアからの利用方法を明らかにしようとする点で独創的である。これに対し従来の Approximate Memory 研究はデバイスレベルの特性を明らかにする最も下層レイヤーの研究および、エラー発生メカニズムを考慮しないソフトウェアレベルでの利用法を明らかにする最も上層レイヤーの研究がほとんどであった。またハードウェアからソフトウェアまですべてのレイヤーを考慮する必要性から挑戦性も高い。例えば研究成果の詳細に記したテーマ A ではハードウェアレベルの動作原理とソフトウェアレベルのメモリレイアウト変換技術両方の知識が必要だった。

5. 主な研究成果リスト

（1）論文（原著論文）発表

1. Soramichi Akiyama, Ryota Shioya. The Granularity Gap Problem: A Hurdle for Applying Approximate Memory to Complex Data Layout. Proceedings of 12th ACM/SPEC International Conference on Performance Engineering (ICPE). 2021. pp. 125 – 132.

（2）特許出願

研究期間累積件数： 0 件

（3）その他の成果（主要な学会発表、受賞、著作物、プレスリリース等）

1. Soramichi Akiyama. Assessing Impact of Data Partitioning for Approximate Memory in C/C++ Code. The 10th Workshop on Systems for Post-Moore Architectures (SPMA). 2020. pp. 1 – 7.
2. 情報処理学会 システム・アーキテクチャ研究会若手奨励賞, 2021 年 7 月.
3. 電子情報通信学会 コンピュータシステム研究専門委員会（CPSY）研究会優秀若手発表賞, 2020 年 12 月.
4. 穐山 空道, 塩谷 亮太. Approximate Memory におけるエラー混入対象データの重要度の事前推定に関する検討. 並列／分散／協調処理に関するサマー・ワークショップ. 2021 年 7 月. pp. 1 – 10.
5. 穐山 空道, 松宮 遼, 吉藤 尚生, 梶 信也. Approximate Memory 制御手法の評価のためのベンチマーク開発. 並列／分散／協調処理に関するサマー・ワークショップ. 2021 年 7 月. pp. 1 – 9.
6. 穐山 空道, 塩谷 亮太. 複雑なデータ構造を持つアプリケーションを対象とした Approximate Memory 適応の検討. Hot Spring Annual Meeting. 2020 年 10 月. pp. 1 – 11.