

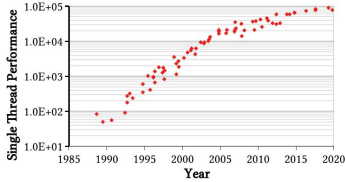
次世代メモリデバイスによるアプリケーションの自動高速化 ~ちょっと間違えて高速に計算~

あきやま 穂山 空道 東京大学 情報理工学系研究科 助教
https://www.soramichi.jp/

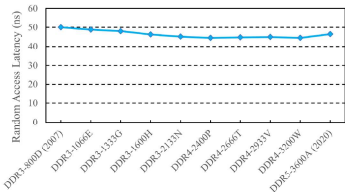
研究の背景と切り拓く未来

解決すべき問題 コンピュータのメモリは遅い

- CPU の計算速度は最近 20 年で 150~200 倍に
- メモリのアクセス遅延はこの間ほぼ不変



1. CPU の単一スレッド性能
データ出典: <https://github.com/karlrupp/microprocessor-trend-data/>



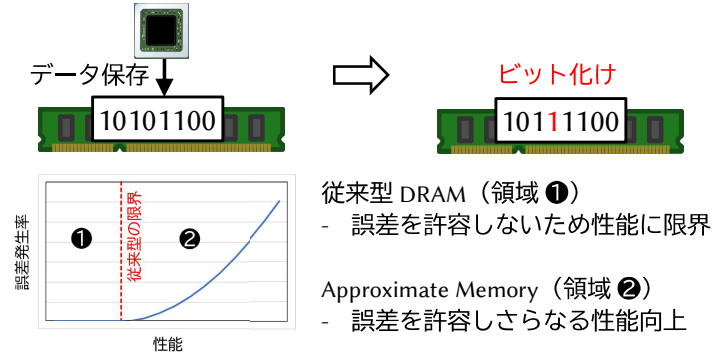
2. メインメモリに使われる DRAM のアクセス遅延*
データ出典: JEDEC 仕様

*あるデータを要求してから、そのデータが出てくるまでの時間

→メモリは相対的に 150 倍以上遅くなっている

切り拓く未来 Approximate Memory の実用化

- 特徴: 高速アクセスできるがデータが化ける

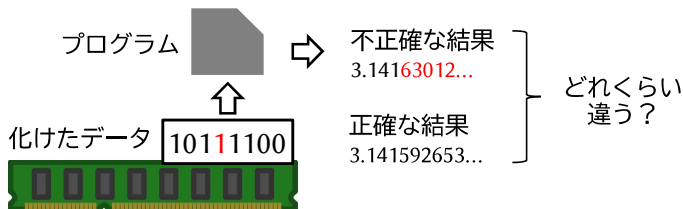


ACT-I 加速期間での成果

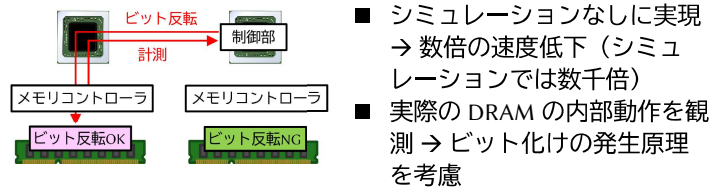
- 通常メモリとの使い分けの困難性の定量評価
- 計算結果の誤差保証への道筋をつけた

研究内容の詳細

ビット化けによる計算結果への影響の見積もり

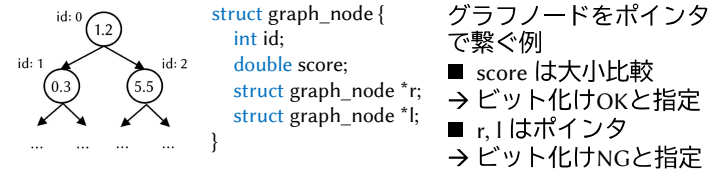


提案: 実機のメモリアクセスを観測しビット化けをモデル化



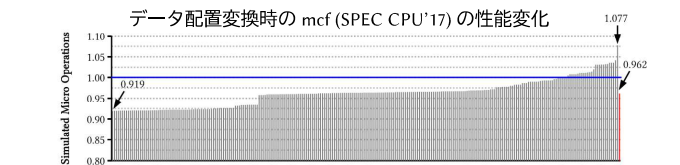
S. Akiyama: "A Lightweight Method to Evaluate Effect of Approximate Memory with Hardware Performance Monitors", *IEICE Trans. Inf. & Syst.*, 2019

通常メモリとの使い分けの困難性の定量評価



発見: 通常のメモリと細粒度な使い分けが必要だがこれは困難

- ビット化けの有無を決める最小単位は 512 B (構造的制約)
- 多くのアプリでそれ以下の単位で異なるデータ型が混同
- メモリ上のデータ配置を無理やり変換すると性能低下



S. Akiyama et al: "The Granularity Gap Problem: A Hurdle for Applying Approximate Memory to Complex Data Layout", *ACM/SPEC ICPE*, 2021

計算結果の許容誤差の保証



従来法 (↑も含む) の欠点

- 制御パラメータを試行錯誤で見つけるしかない
- どんな入力データでも誤差が希望の値以下になる保証がない

提案: プログラムの微分値を用いた誤差保証

- 微分値: 入力 x が dx だけずれると、出力 y は dy ずれる → 誤差を dy に保証するには入力は dx しかずれてはいけない

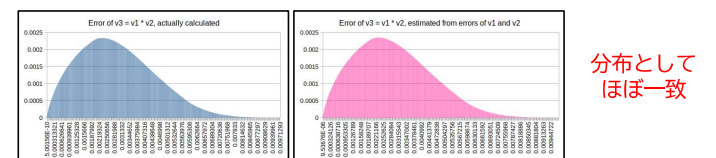
$$x + dx \rightarrow F \rightarrow y + dy \quad dx \leftrightarrow \frac{\partial F}{\partial x} \leftrightarrow dy$$

- 微分値の入力データへの感度を調べた (IPSJ SIGARC 若手賞)

今後の展望

1. 微分値による誤差保証と誤差の分布の組み合わせ
 - 計算誤差の最大値だけでなく計算誤差の分布が知りたい
 - 進行中の研究: 入力変数の誤差を確率変数だと思い、微分による誤差伝播モデルを適用

v3 = v1 * v2 を小数部 8 bit の固定小数点で計算した時の計算誤差
左: 実際に計算 右: v1, v2 の誤差分布を入力に提案手法を適用



2. 新たな脆弱性 (電気的不安定性、attack surface の増加) の考慮