# リソース分離型マイクロデータセンターにおけるリモートメモリ活用手法の性能評価と考察

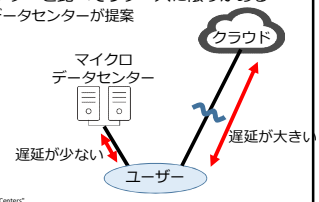
大阪大学　大学院情報科学研究科　村田研究室
生駒 昭繁



## マイクロデータセンターの登場

- クラウドによる処理の限界
  - 遠方のデータセンターとの通信による遅延
  - リアルタイム性の高い処理には不向き
- マイクロデータセンターの活用
  - エッジに小規模のデータセンターを配置
    - 通信遅延の削減
  - しかし、大規模なデータセンターと比べてリソースに限りがある
    - リソース分離型マイクロデータセンターが提案

マイクロデータセンター

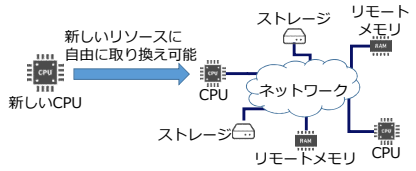



出典：“Vapor IO Forms Alliance for Full-Service Edge Computing at Its Tower Data Centers”
https://www.datacenterknowledge.com/vapor-io/vapor-io-forms-alliance-full-service-edge-computing-its-tower-data-centers, last accessed on 2020-2-16.



## リソース分離型マイクロデータセンター

- リソース単位で構成されたマイクロデータセンター
  - 効率のよいリソース利用が可能
    - リソースごとの最適化が可能
  - 他のリソースを考えずにリソースの追加、変更が可能
- 処理ごとに各リソースの資源を割り当て可能
  - リソースを無駄なく割り当てることができる
  - リソース使用率の向上

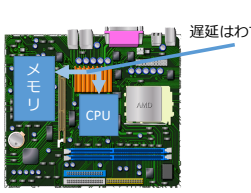



リソース分離型マイクロデータセンター

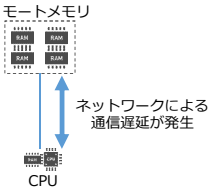


## リソース分離によって発生する課題

- マザーボード上のリソースで構成されるサーバと比べて性能低下の恐れ
  - リソース間での通信による遅延が発生
- CPU とメモリ間の通信遅延は実行時間に大きく影響[5]
  - 性能低下の直接の原因
  - 遅延の影響を少なくするための取り組みが必要




マザーボード上のやり取り




リモートメモリとのやり取り

[5] P. X. Gao, A. Narayan, S. Karandikar, J. Carreira, S. Han, R. Agarwal, S. Ratnasamy, and S. Shenker, “Network requirements for resource disaggregation,” in 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 249–264, Nov. 2016.



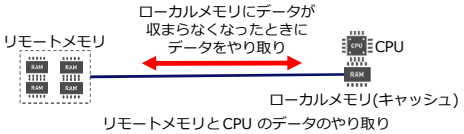
## 研究目的とアプローチ

- 研究の目的
  - リソース分離型マイクロデータセンターにおける
    リモートメモリ活用時の性能評価
- アプローチ
  - リモートメモリと CPU 間のやり取り時にネットワーク遅延を挿入することでリモートメモリとの通信を再現
  - リモートメモリとの通信遅延を挿入した状態でエッジでの実行が想定される処理を実行し、実行時間による性能低下率を導出
  - 性能低下率をもとにリモートメモリの活用方法について考察



## リモートメモリと CPU のデータのやり取り

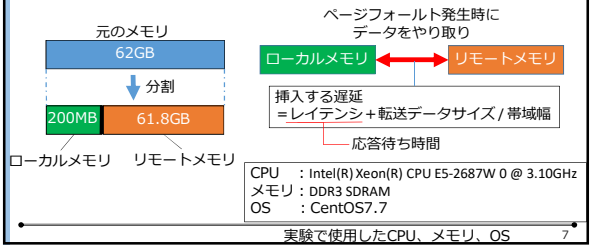
- リモートメモリはローカルメモリを介して CPU とやり取り
  - ローカルメモリはリモートメモリのキャッシュメモリとして動作
- ローカルメモリと CPU は直接接続
  - ローカルメモリと CPU との通信遅延はわずか
- ローカルメモリに処理データが収まらなくなったときにリモートメモリと通信
  - リモートメモリをローカルメモリの追加領域として利用
- ページ単位でデータをやり取り
  - ページサイズは 4KB に固定




リモートメモリと CPU のデータのやり取り

## 評価環境の構築

- ローカルメモリとリモートメモリの通信時に遅延を挿入することで CPU との通信時のネットワーク遅延を再現
  - リモートメモリはローカルメモリを通して CPU と通信するため
- 元のメモリをローカルメモリとリモートメモリに分割
- 挿入する遅延はレイテンシと帯域幅をもとに決定
  - それぞれパラメータとして値を指定可能




CPU　：Intel(R) Xeon(R) CPU E5-2687W 0 @ 3.10GHz
メモリ：DDR3 SDRAM
OS　：CentOS7.7

実験で使用したCPU、メモリ、OS



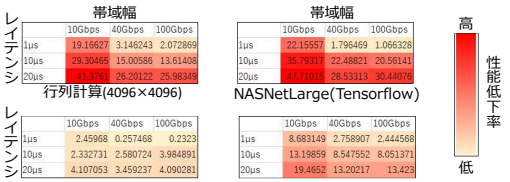
## 評価方法

- 行列計算とその応用である機械学習による画像分類を実行することで性能を評価
  - 行列計算：4096×4096 の行列と 2048×2048 の行列に対する畳み込み計算処理
  - 画像分類：100 個の画像に対する画像分類処理
    - 7 個の機械学習モデルを Tensorflow と Tensorflow Lite の 2 つのライブラリで実行し、計 14 種類の処理について評価
    - 畳み込みニューラルネットワークのモデルを使用
- 評価対象を 10 回実行し、その実行時間の中央値を測定
- 測定値をもとに性能低下率を導出
- レイテンシと帯域幅を変更しながら評価対象と性能低下率の関係を評価
  - レイテンシ：1μs、10μs、20μs に変更
  - 帯域幅：10Gbps、40Gbps、100Gbps に変更



## 評価結果

- 評価対象によって性能低下率は様々
  - リモートメモリアクセスによる遅延の影響を受けやすい処理と受けにくい処理が存在
  - 帯域幅とレイテンシと性能低下率の相関がみられない評価対象も存在
- 帯域幅よりもレイテンシの影響を強く受ける
- 40Gbps 以上の帯域幅にしても大きな性能の改善は見込めない

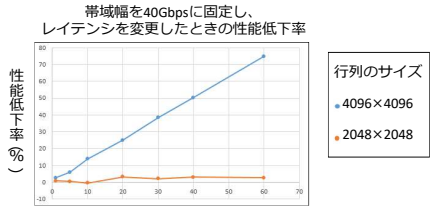



評価対象ごとのレイテンシと帯域幅を変更したときの性能低下率（一部抜粋）



## 行列サイズと性能低下率の関係

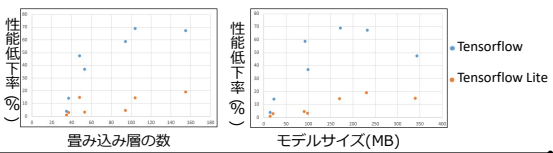
- 行列のサイズが大きい程性能低下率は増加
  - 計算時の計算量が増加するため
- 2048×2048 の行列に対する畳み込み計算は性能低下率にほとんど変化がなかった
  - 行列サイズが小さければ、レイテンシや帯域幅の影響を受けにくい






## 行列計算と機械学習モデルの関係

- モデル内の畳み込み層が多いほど性能低下率が高い傾向
  - 行列計算(畳み込み計算)は遅延の影響を強く受けるため
  - 両方のライブラリで正の相関がみられる
    - 相関係数は Tensorflow で 0.830466 、Tensorflow Lite で 0.717489
- モデルサイズが大きいほど性能低下率が高い傾向
  - 相関係数は Tensorflow で 0.649346、Tensorflow Lite で 0.846798
    - Tensorflow Lite では、畳み込み層の数よりも相関が高い
    - Tensorflow Lite は計算時のパラメータ数を削減するので、畳み込み計算による性能低下の影響が小さくなるためと考えられる




性能低下率と畳み込み層の数とモデルサイズの関係



## リモートメモリの活用についての考察

- レイテンシは性能の低下に大きく影響を及ぼす

↓

- レイテンシを考慮したリモートメモリの配置が必要
  - レイテンシを 5μs 以下に抑えたい場合：
    - CPUから少なくとも 400m 以内にはリモートメモリを配置
    - 伝搬速度が 200m/μs であるため
- 遅延の影響を受けやすい処理と受けにくい処理が存在

↓

- リモートメモリの割り当てアルゴリズムが必要
  - 遅延の影響を受けやすい処理には近くに配置されたリモートメモリを割り当てる必要
  - リモートメモリアクセスの多い処理を判別するための指標が必要
    - 例：機器学習モデルの畳み込み層の数、行列のサイズ

## まとめと今後の課題

- まとめ
  - リモートメモリ活用時の性能の調査を実施
  - リモートメモリの活用による遅延の影響が大きい処理と小さい処理が存在
  - 様々な処理でリモートメモリを用いることができるような配置や割り当て方法が必要
- 今後の課題
  - リソースの分離を前提にしたアルゴリズムによる効果の検討
  - データ転送による遅延の影響を減らす対策の検討
    - データの圧縮による転送遅延の削減
    - 効率的なデータ転送スケジューリング
    - 最適なページサイズ
  - ネットワークの輻輳による影響の調査

13