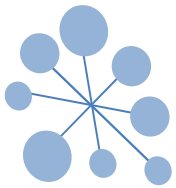


仮想I/Oのキャッシュ汚染解消に向けた Non-Temporal命令の有効性評価

○竹谷 大地[†] 川島 龍太[†]
中山 裕貴^{††} 林 經正^{††} 松尾 啓志[†]

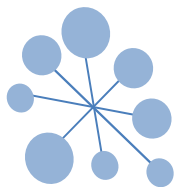
[†] 名古屋工業大学

^{††} 株式会社ボスコ・テクノロジーズ



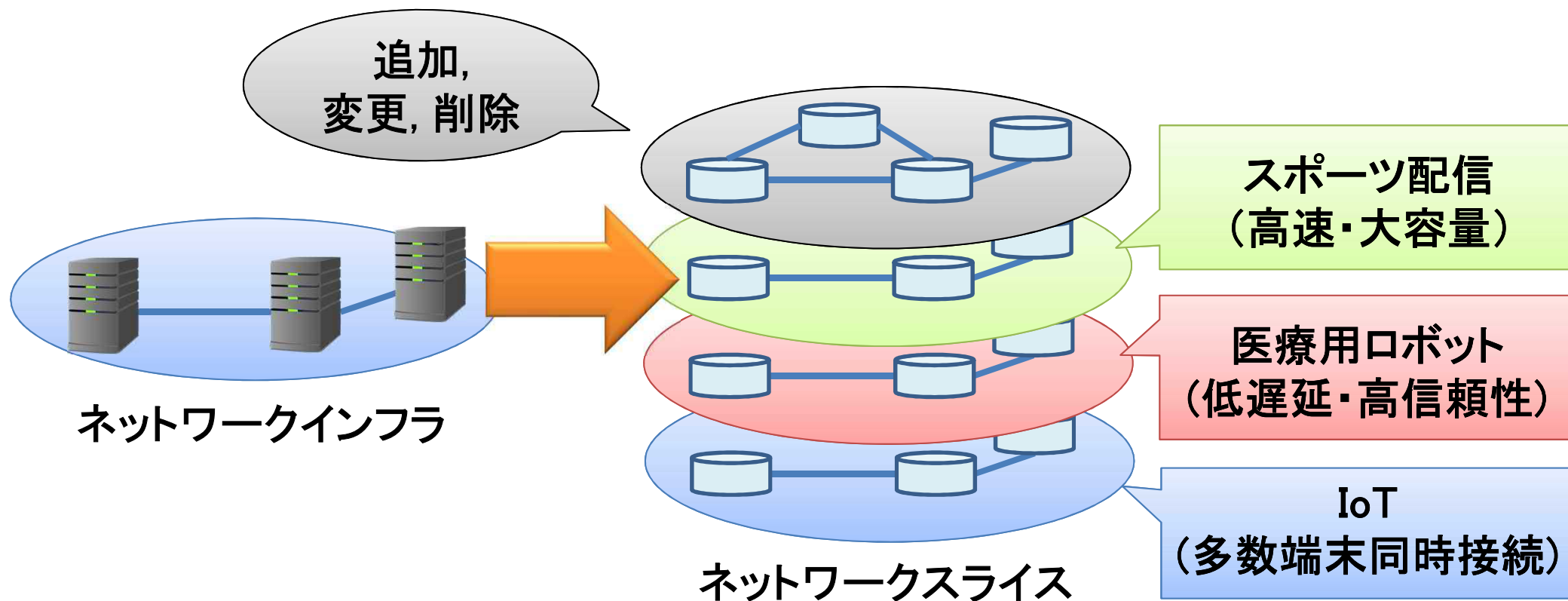
目次

- 研究背景
- 関連研究
- 現状の問題点整理とNon-temporal命令
- 評価
 - キャッシュヒットレート
 - LFBによるストール
 - 読み取り時のメモリアクセスへの影響

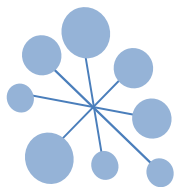


5Gネットワークの要求

- 多様なネットワークの迅速かつ柔軟な展開
 - 即時の機能追加, 変更, 削除が可能

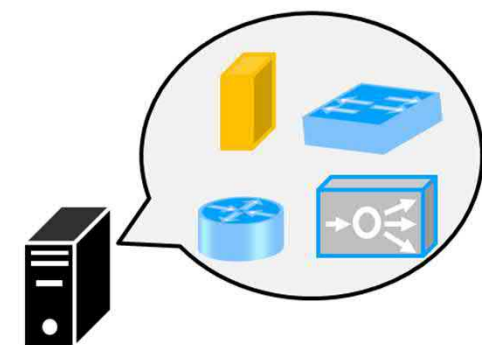


ネットワーク機能の仮想化 (NFV) が必須となる

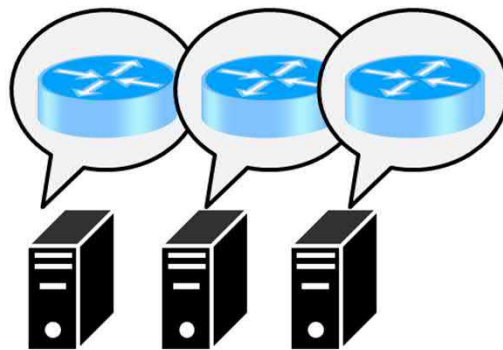


CNF (Cloud Native Network Function)

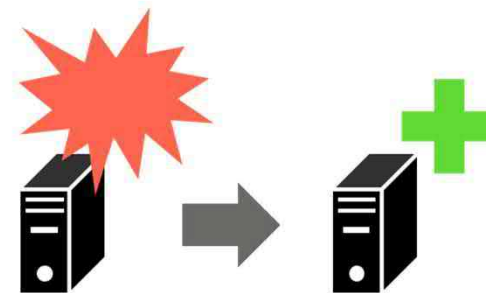
- クラウドネイティブなNF実装
 - Kubernetesによる運用の自動化が可能



オートデプロイメント



オートスケーリング

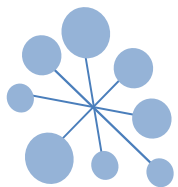


オートヒーリング

パケット処理能力は10Gbps程度

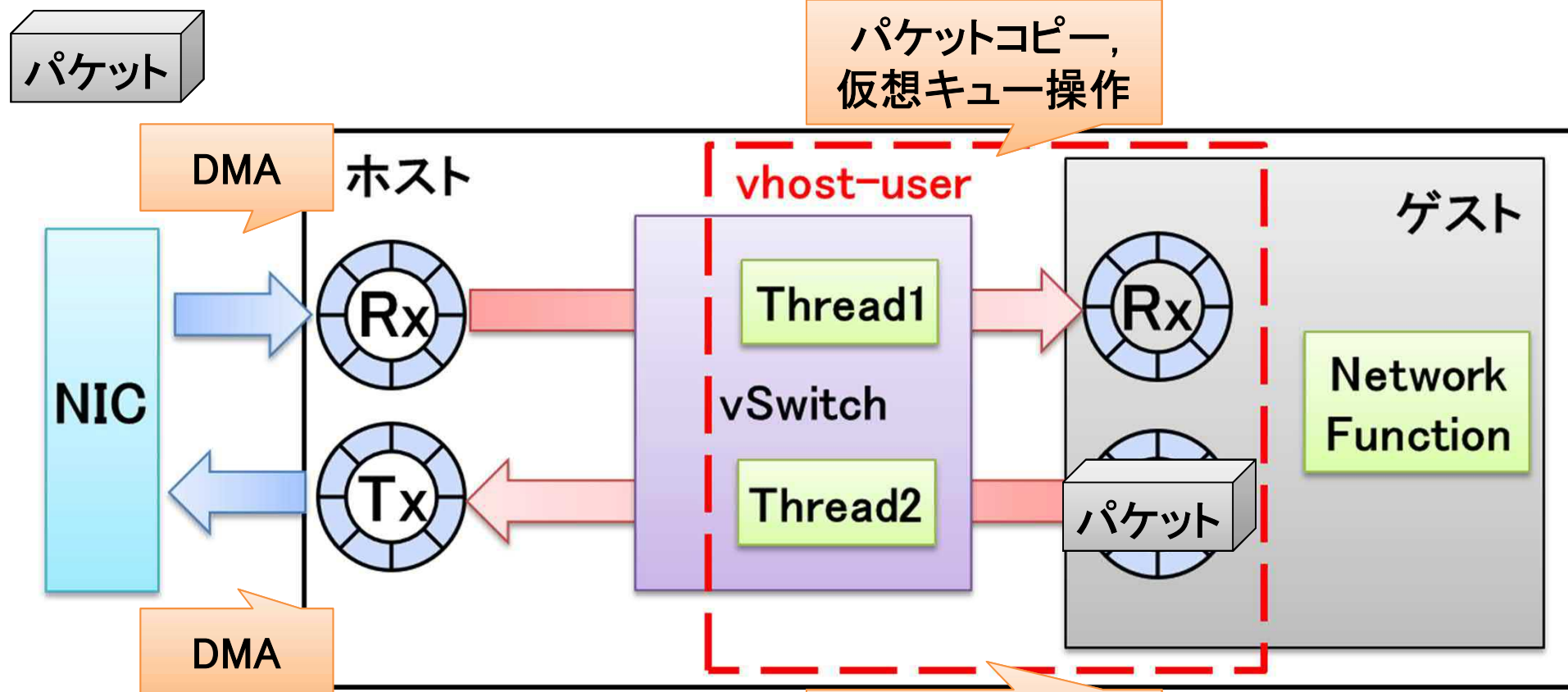


コンテナ環境のパケットI/O高速化が必要

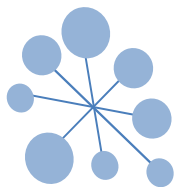


仮想環境におけるパケット処理

- vhost-user
 - ユーザ空間上でプロセス間通信を行う仮想I/Oプロトコル

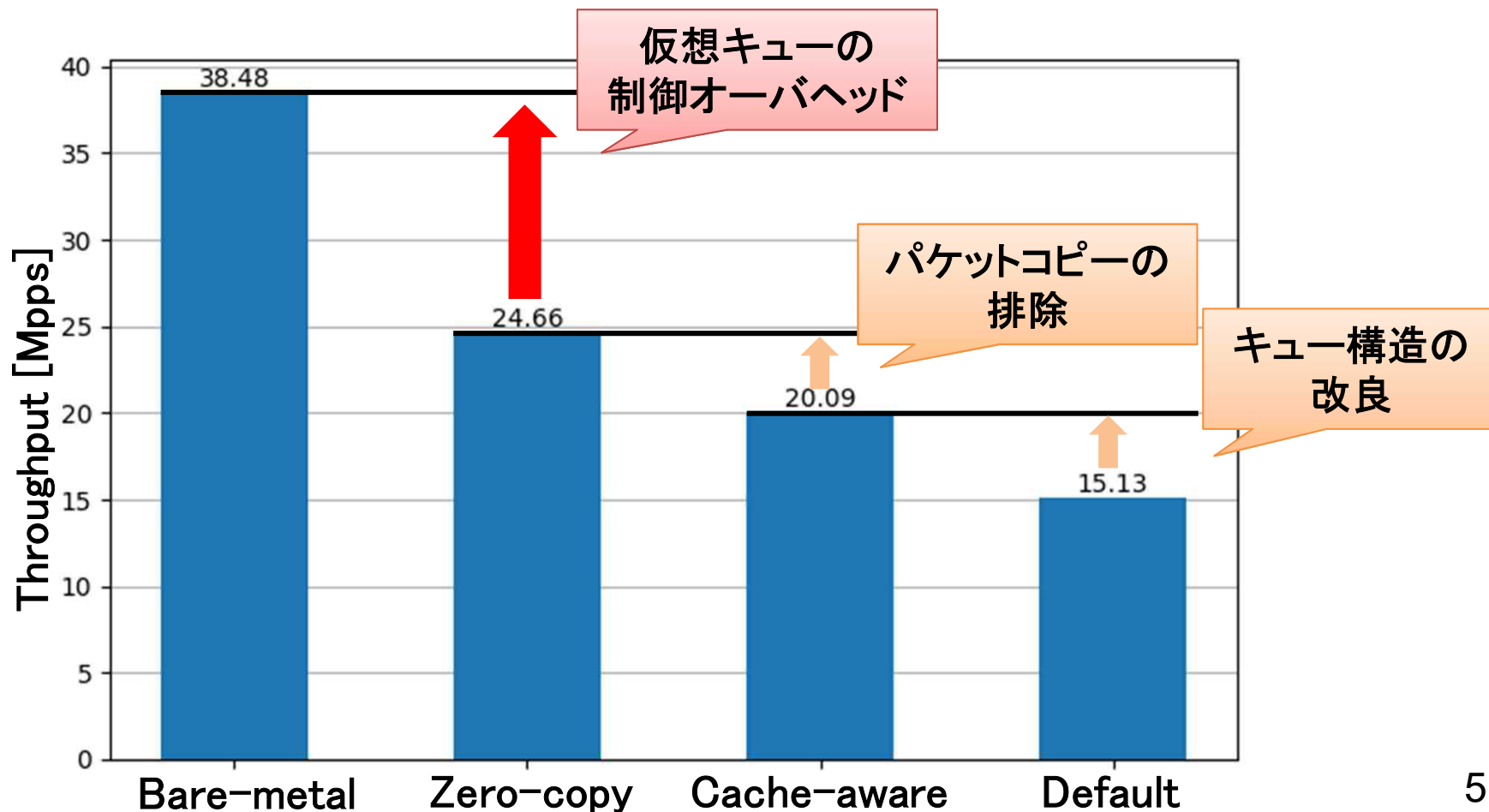


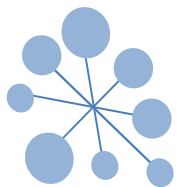
パケットコピーと仮想キュー操作がボトルネックに



仮想I/O性能の推移

- 主な仮想I/Oの性能向上技術
 - キャッシュ利用効率に優れた仮想キュー構造
 - パケットコピーの排除





仮想キュー操作の高速化

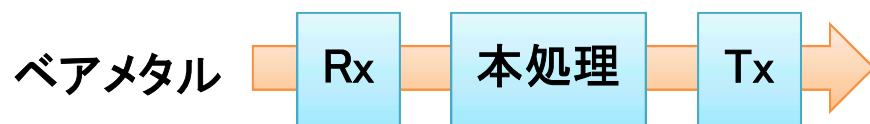
- パケットあたりの実行時間の削減が必要

- 処理量を減らす
- 処理中の待ち時間を減らす

処理量の削減は難しい



メモリアクセスの待ち時間の削減が必要



- 暗黙的なキャッシュ制御

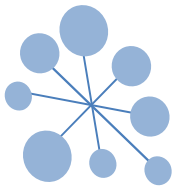
- プリフェッチを想定したアクセス順序
- メモリアライメントを揃える
- キャッシュラインを意識したデータサイズ

既に最適化済み

- 明示的なキャッシュ制御

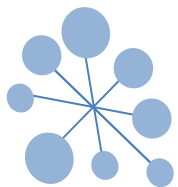
- キャッシュの動きを直接制御
- 変数にキャッシュ領域を割り当て

まだ実装されていない



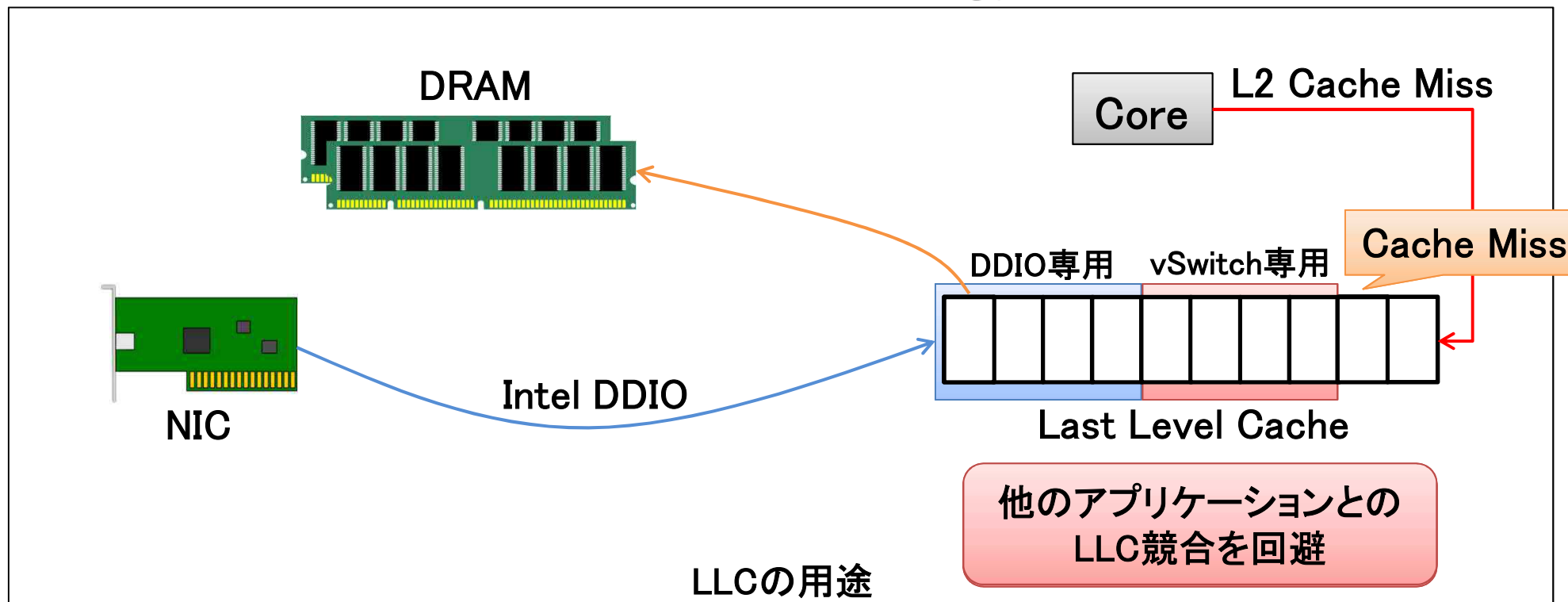
目次

- 研究背景
- 関連研究
- 現状の問題点整理とNon-temporal命令
- 評価
 - キャッシュヒットレート
 - LFBによるストール
 - 読み取り時のメモリアクセスへの影響



Last Level Cacheの明示的制御

- アプリケーションごとにLLCを排他的に割り当て
 - Intel Cache Allocation Technologyを使用



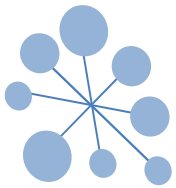
L1, L2キャッシュに対して適用できない

† A. Farshin, A. Roozbeh, G.Q.M. Jr., and D. Kostić,

“Reexamining direct cache access to optimize i/o intensive applications for multi-hundred-gigabit networks”, USENIX ATC 20

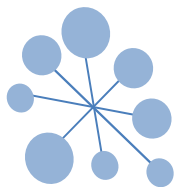
†† S. Thomas, R. McGuinness, G.M. Voelker, and G. Porter,

“Dark packets and the end of network scaling”, ANCS '18



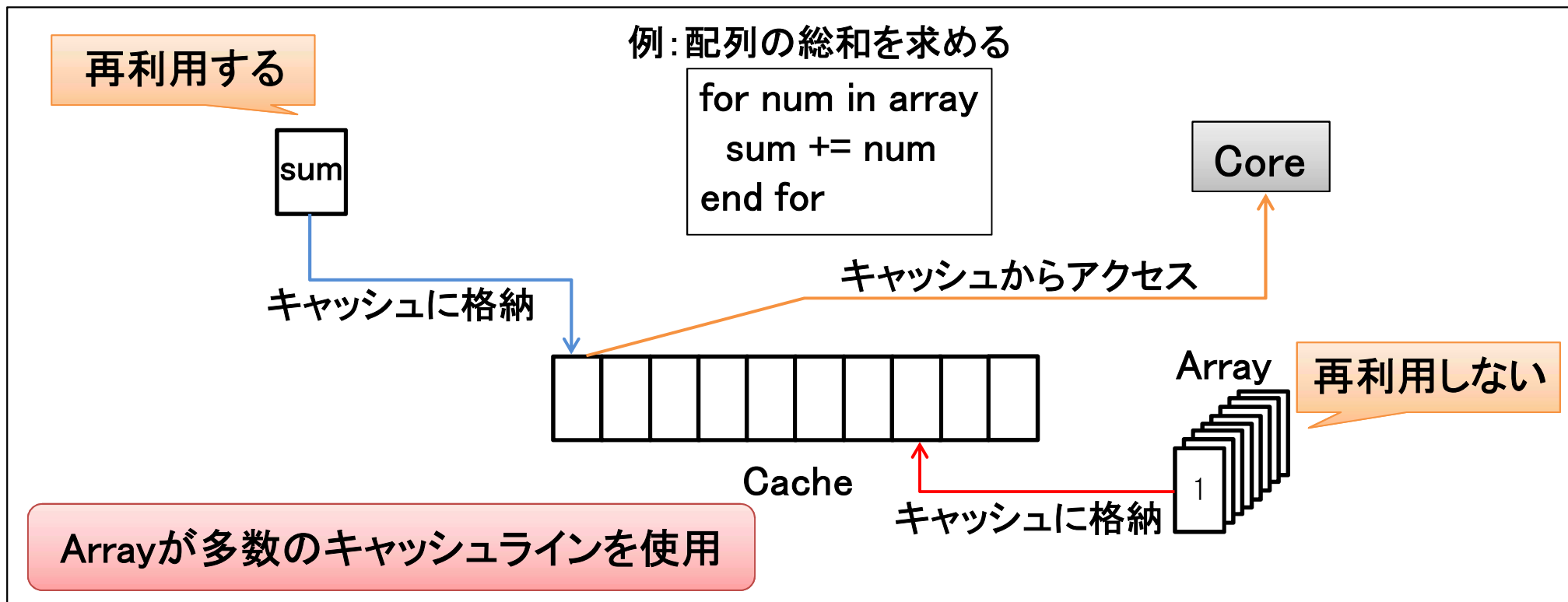
目次

- 研究背景
- 関連研究
- **現状の問題点とNon-temporal命令**
- 評価
 - キャッシュヒットレート
 - LFBによるストール
 - 読み取り時のメモリアクセスへの影響



キャッシュ汚染

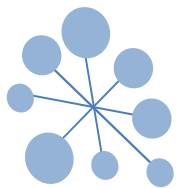
- 再利用しないデータがキャッシュの大部分を使用



暗黙的なキャッシュ制御では解消できない

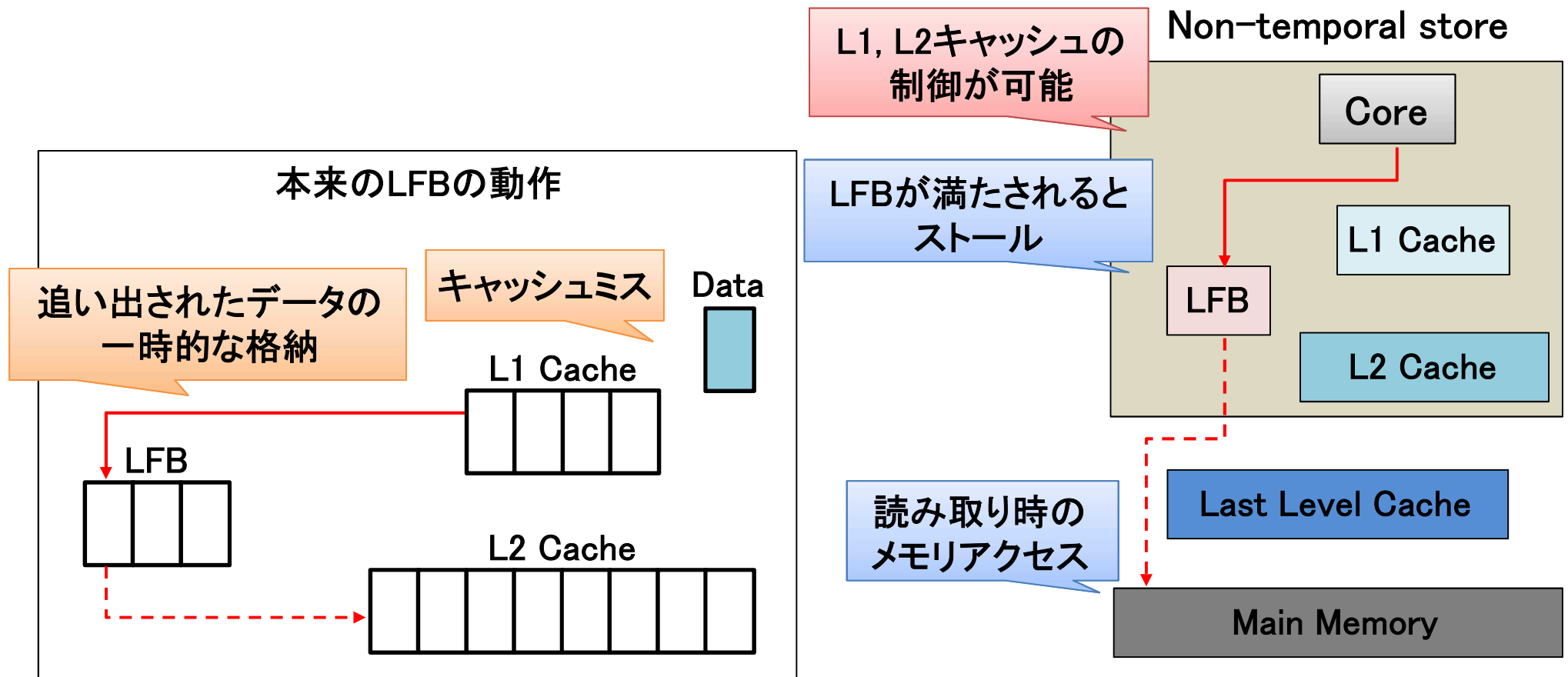


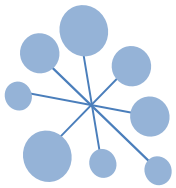
再利用しないデータを残さない仕組みが必要



Non-temporal store命令

- キャッシュを一切利用しないストア命令
 - Line Fill Buffer (LFB) を使用





仮想I/O高速化の要件

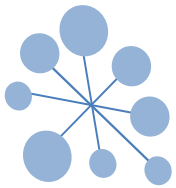
- キャッシュ制御によるキャッシュの利用効率向上
- キャッシュ汚染の解消が必要
 - 特にL1, L2キャッシュのヒット率を向上させたい
 - キャッシュを利用しないNon-temporal命令を使用

現段階では適用すべき箇所がわからない

使用上のデメリットも存在する



まずはNon-temporal命令の有効性を調査



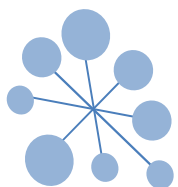
Non-temporal命令の有効性評価

- Non-temporal命令による有効性を評価
 1. キャッシュヒットレートの向上するか
 - キャッシュ汚染の解消が実現できているか
 2. LFBのフラッシュが効果的に行えているか
 - LFBのエントリが全て満たされると同期的にメモリへ書き込む
 3. 読み取り時のメモリアクセスによる性能への影響

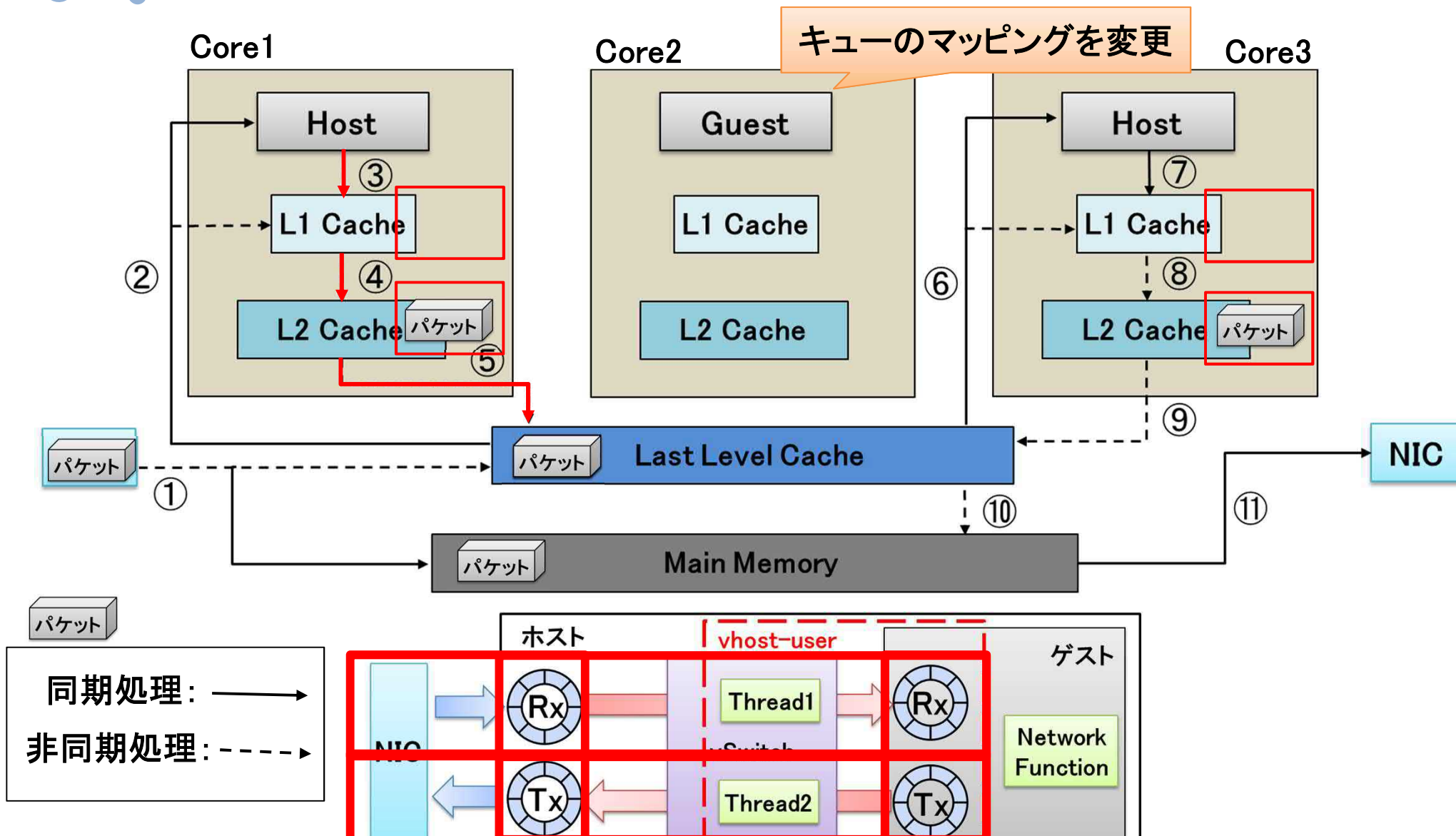
効果がわかりやすい箇所に適用して評価したい



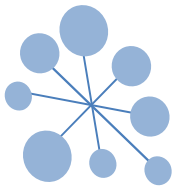
パケットコピーに対して適用



パケット処理におけるキャッシュ利用

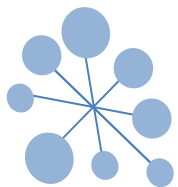


ホストからゲスト空間に対して発生するパケットコピーに適用



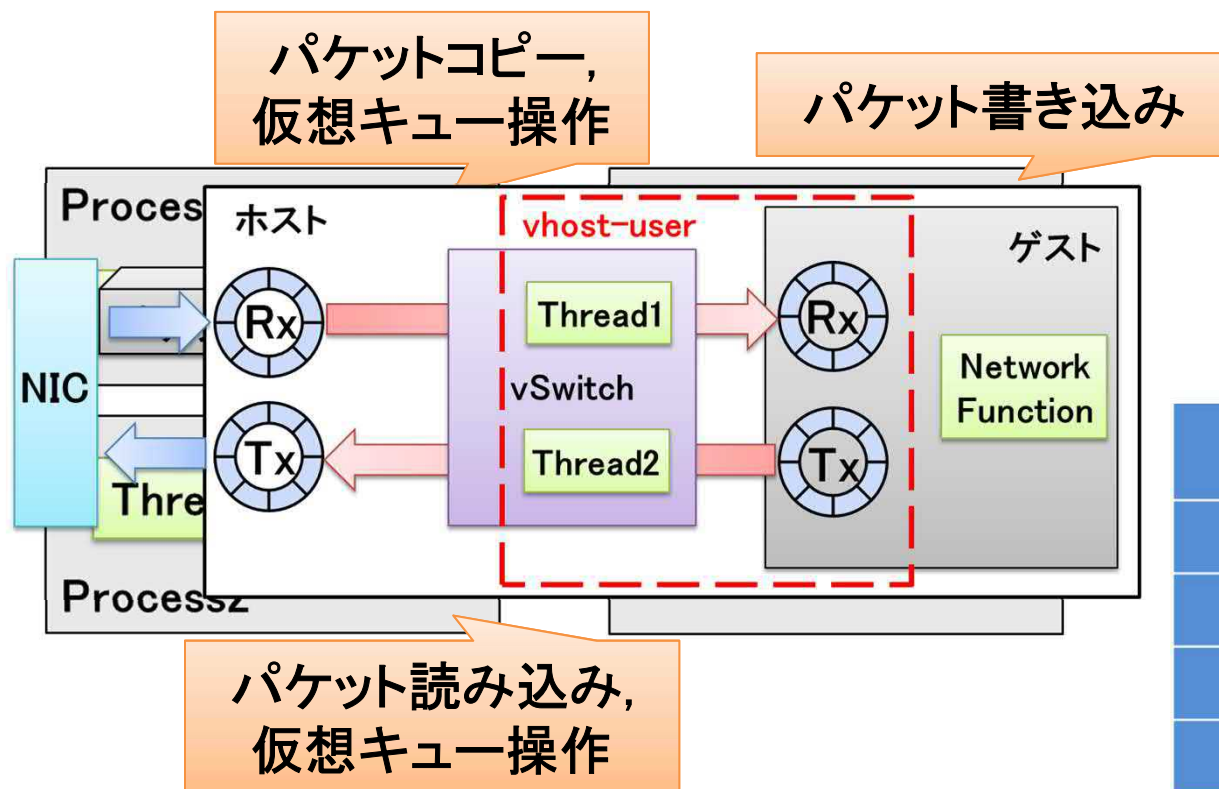
目次

- 研究背景
- 関連研究
- 現状の問題点とNon-temporal命令
- 評価
 - キャッシュヒットレート
 - LFBによるストール
 - 読み取り時のメモリアクセスへの影響

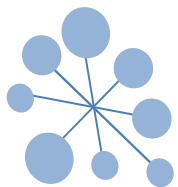


評価環境

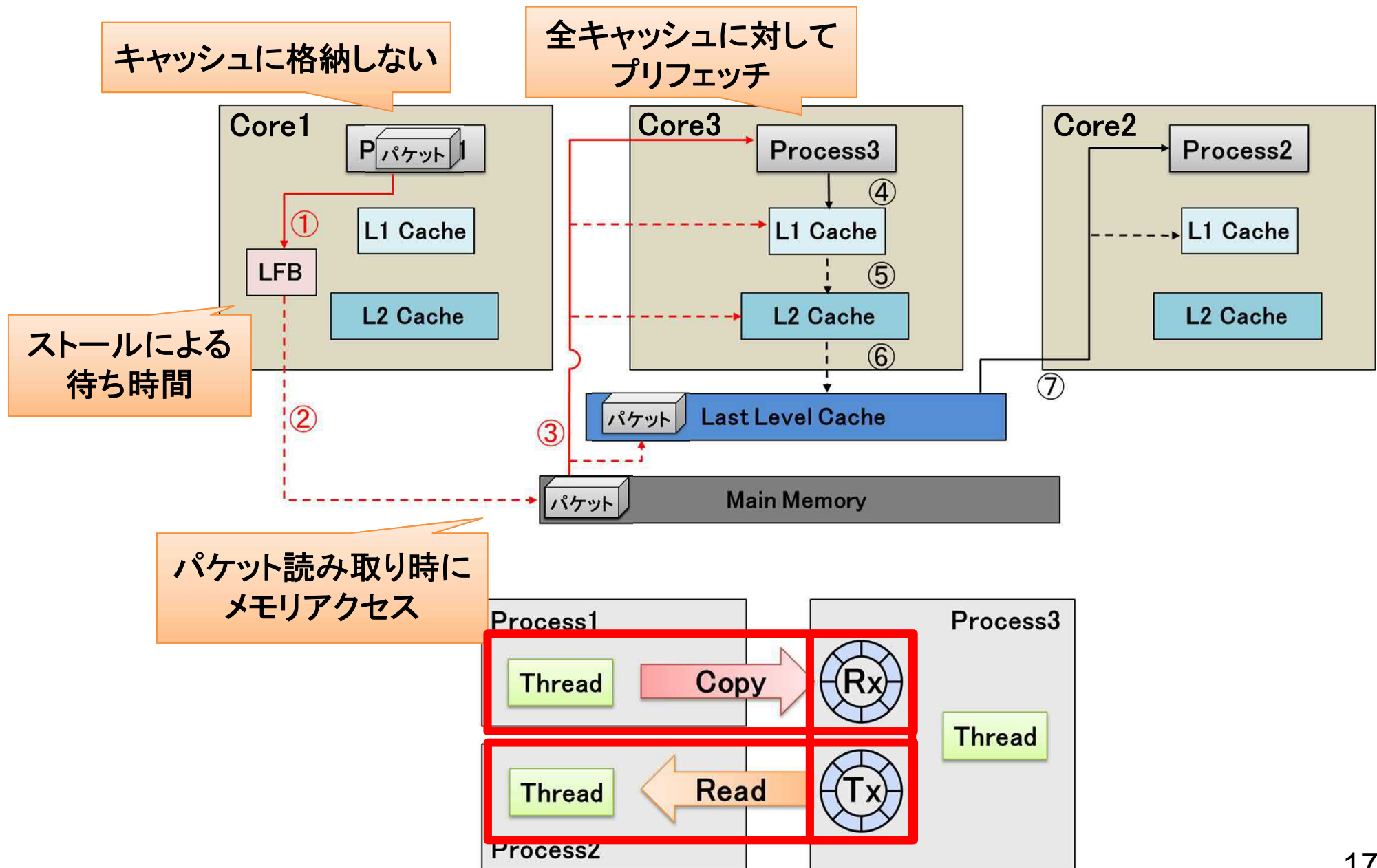
- パケットコピーとキュー操作に限定したプログラム
 - 本質的なパケット転送に関係ない処理の影響を回避
 - オフロード, ルーティング, etc...

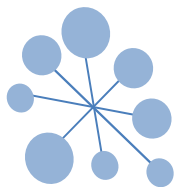


CPU	Intel Core i7-6900K 3.20GHz 8コア (HT無効)
L1i Cache	32KB 8-way
L1d Cache	32KB 8-way
LFB Entries	10 cache lines
L2 Cache	256KB 8-way
Last Level Cache	20480KB 20-way



評価プログラムの流れ





キャッシュヒットレートの推移

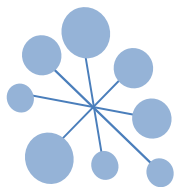
L1キャッシュミスの削減

64 bytes	Temporal store	Non-temporal store
L1 Miss	75.3M	66.9M
L1 Miss Ratio	9.3%	8.5%
L2 Miss Ratio	66.8%	67.9%

パケットサイズの増加に伴ってより効果的に

1024 bytes	Temporal store	Non-temporal store
L1 Miss	323.7M	112.0M
L1 Miss Ratio	15.2%	9.3%
L2 Miss Ratio	12.2%	6.9%

パケットデータによるキャッシュ汚染が解消できている



LFBのストール待ちへの影響評価

- LFBのエントリが満たされてしまうとストールが発生
 - 書き込みの間隔による性能への影響を調査

```
/* パケットコピーと仮想キュー更新が別ループ */  
for each packet of batch do  
    copy_packet ← パケットコピー  
end for  
for each packet of batch do  
    update_virtqueue ← 仮想キュー更新  
end for
```

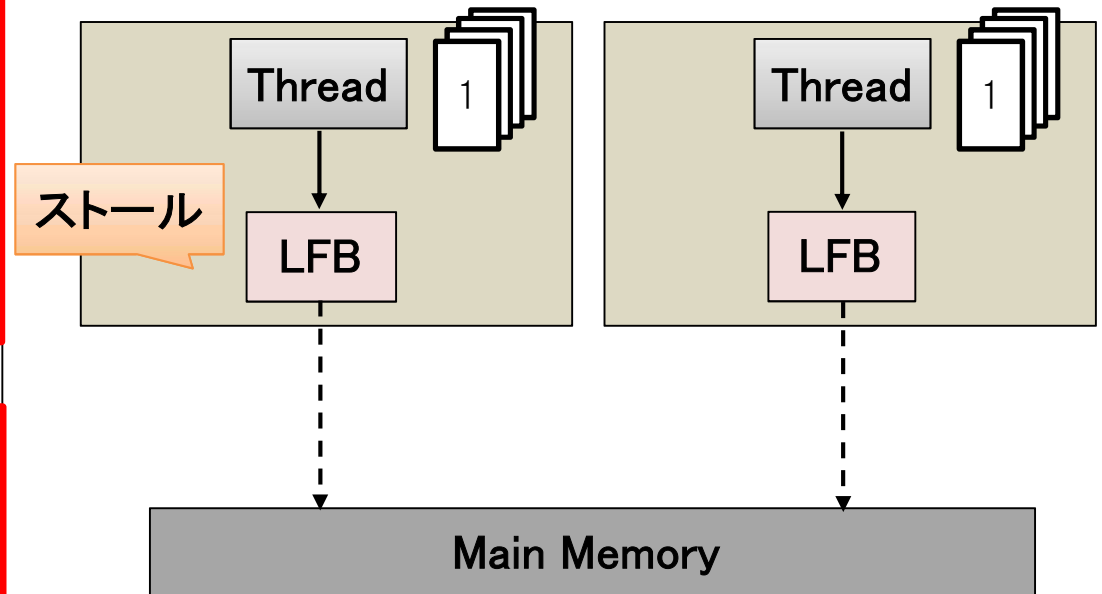
```
/* パケットコピーと仮想キュー更新が同一ループ */  
for each packet of batch do  
    copy_packet ← パケットコピー  
    update_virtqueue ← 仮想キュー更新  
end for
```

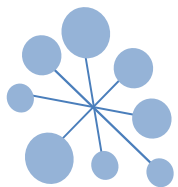
LFBの使用率を急激に高めてしまう

メモリに書き戻す時| キュー操作

バッチ数分ストアした後にバッチ数分キュー操作

コピー後に必ずキュー操作





LFBのストール数と待ち時間の推移

Non-temporal命令での評価

5%の性能差

64 bytes	連続でコピー	間隔をあけてコピー
LFB Miss	256.5M	203.4M
LFB Pending (cycles)	4300.2M	4173.5M
Throughput	42.5Mpps	45.3Mpps

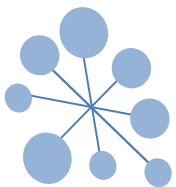
待ち時間は増加せず

1024 bytes	連続でコピー	間隔をあけてコピー
LFB Miss	823.7M	701.4M
LFB Pending (cycles)	1302.3M	1410.2M
Throughput	9.2Mpps	9.1Mpps

LFBによるストール数は向上するが待ち時間はそれほど増加しない



LFBによるストールは顕著な性能低下を引き起こしていない



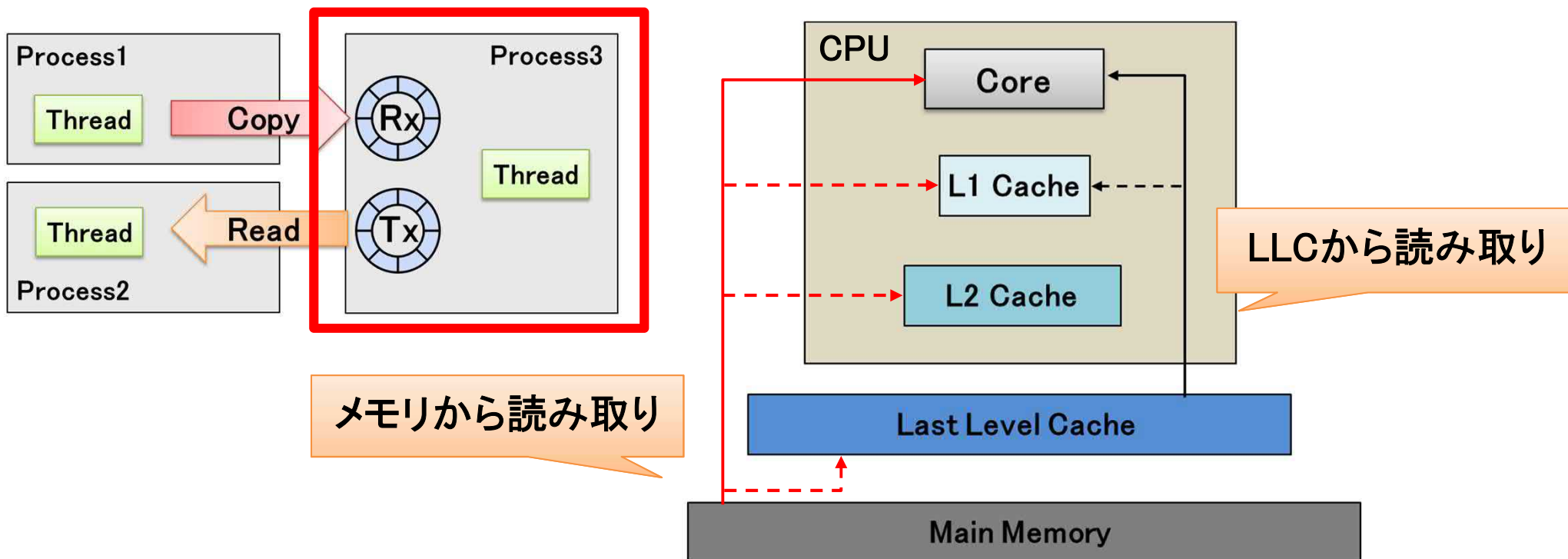
受信時のメモリアクセスの影響評価

Temporal Store (同期): —→

Non-Temporal Store (同期): —→

Temporal Store (非同期): - - ->

Non-Temporal Store (非同期): - - ->



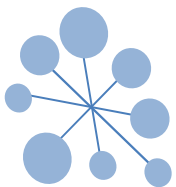
メモリから読み取り

LLCから読み取り

メモリアクセスを無くした状態の性能と比較



メモリアクセスがどの程度性能に影響を与えているか調査



評価環境

Temporal Store (同期): \longrightarrow

Non-Temporal Store (同期): \longrightarrow

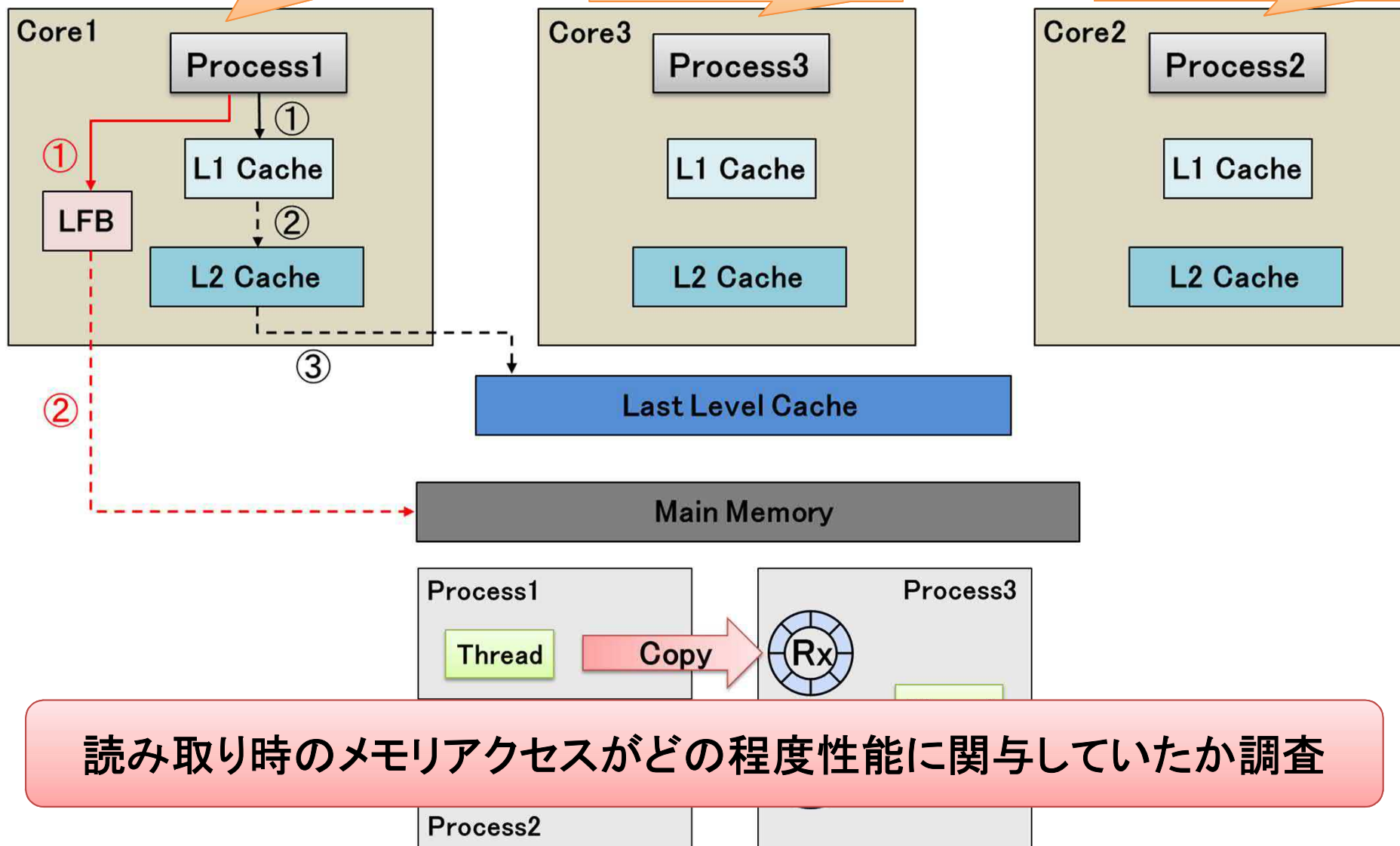
Temporal Store (非同期): \dashrightarrow

Non-Temporal Store (非同期): \dashrightarrow

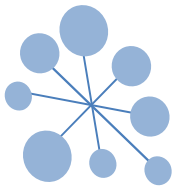
パケットコピー,
キュー操作

キュー操作のみ

キュー操作のみ

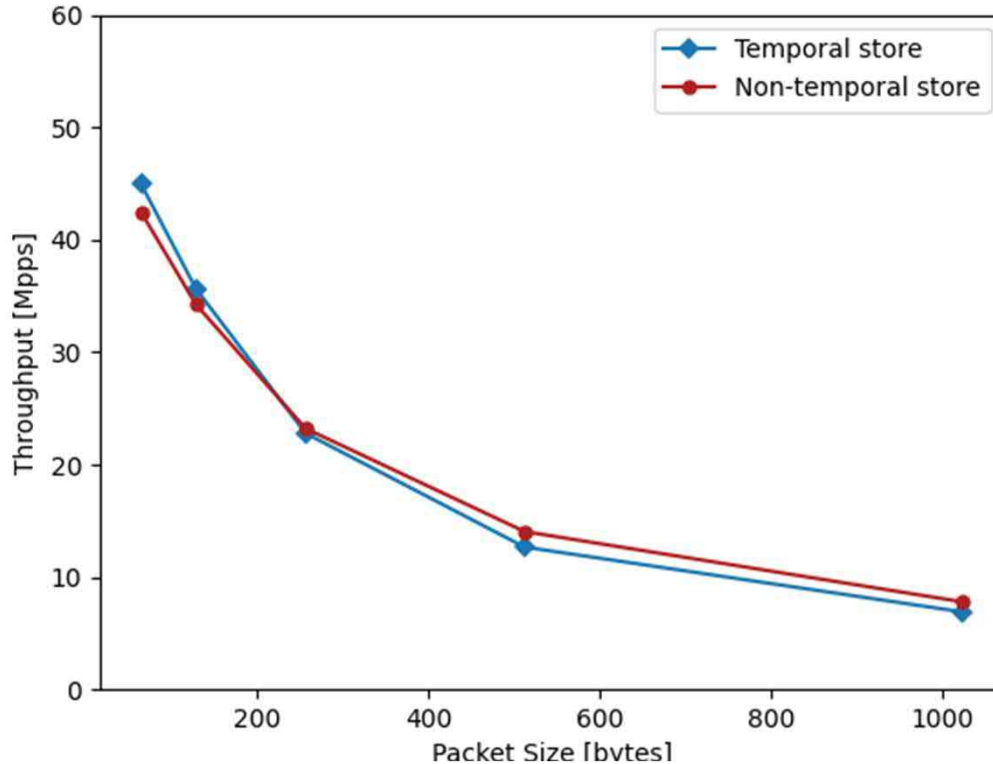


読み取り時のメモリアクセスがどの程度性能に關与していたか調査

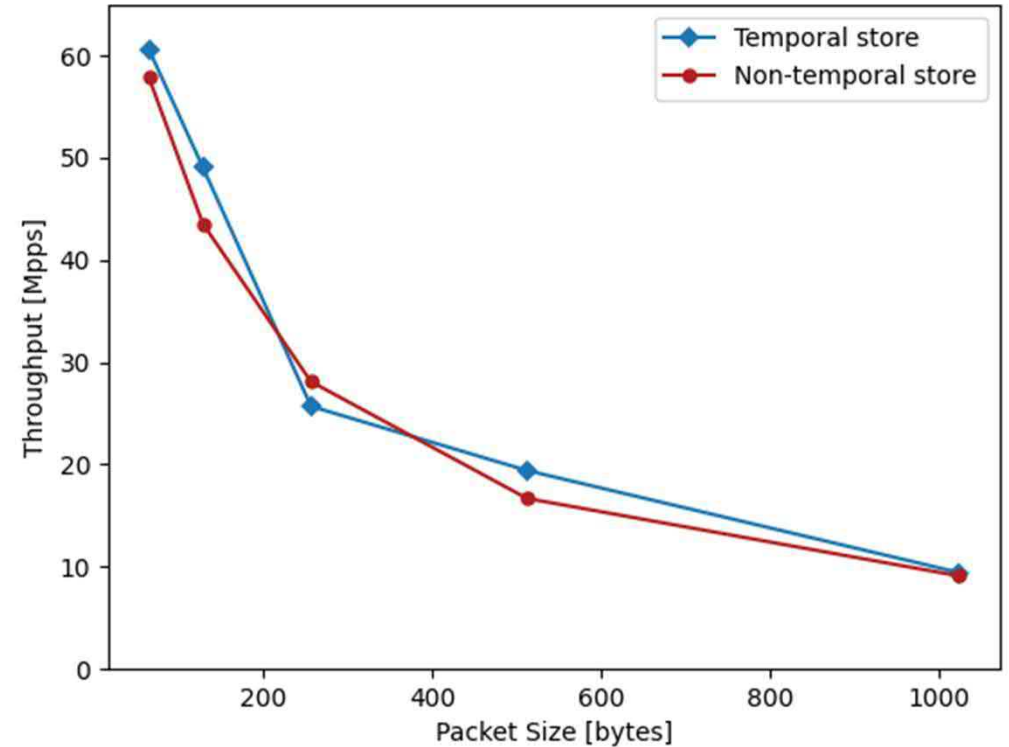


スループット比較

メモリアクセスあり



メモリアクセスなし

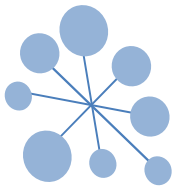


読み取り時のメモリアクセスによる顕著な性能低下は見られない



キャッシュ利用効率の向上が
他の要因と相殺

ボトルネックは他の箇所があり
そこが改善されていない



まとめと今後の課題

- Non-temporal命令による有効性を評価
 - L1キャッシュヒットレートが向上
 - デメリットとなる要因による顕著な性能低下は見られなかった
 - LFBによるストール
 - 読み取り時のメモリアクセス

- 今後の課題
 - ボトルネックの特定
 - 仮想キュー操作へのNon-temporal命令の適用