Analyzing Performance Pitfalls of On-Demand Paging of InfiniBand

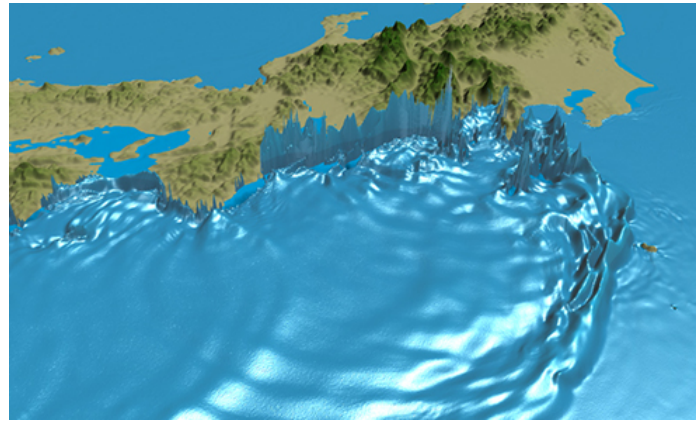
SWoPP 2020 (July 31)

Takuya Fukuoka, Shigeyuki Sato, and Kenjiro Taura

The University of Tokyo

マルチノードのスパコンシステム

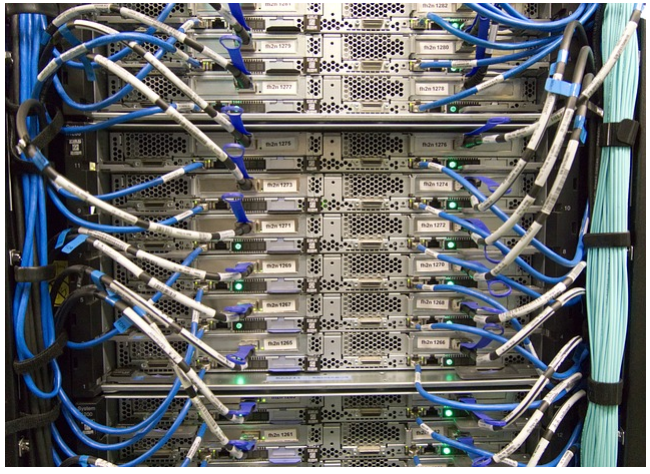
- スパコンは科学技術計算において不可欠
- TOP500にランクインするスパコンのほとんどはマルチノードで構成（2020年6月発表）
  - Rank 1. 富岳 : 158,976ノード
  - Rank 2. サミット : 4,608ノード
  - Rank 3. シエラ : 4,474ノード

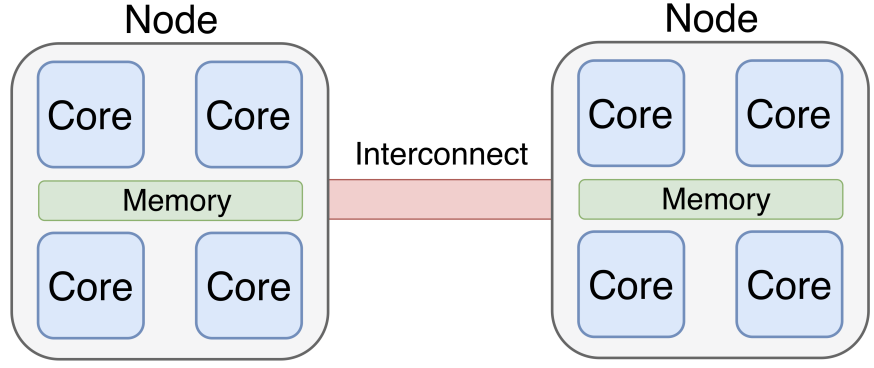


https://www.nikkei.com/article/DGXMZO60655390S0A620C2MM8000/
https://www.itmedia.co.jp/enterprise/articles/1711/27/news048.html

インターコネクトの重要性

- ノード間通信はインターコネクトを通して実行
  - InfiniBand, Ethernet, Omni-Path, Tofuなど
- 全体性能はインターコネクトの性能に大きく影響される
  - 性能は計算よりも通信に律速 [1]



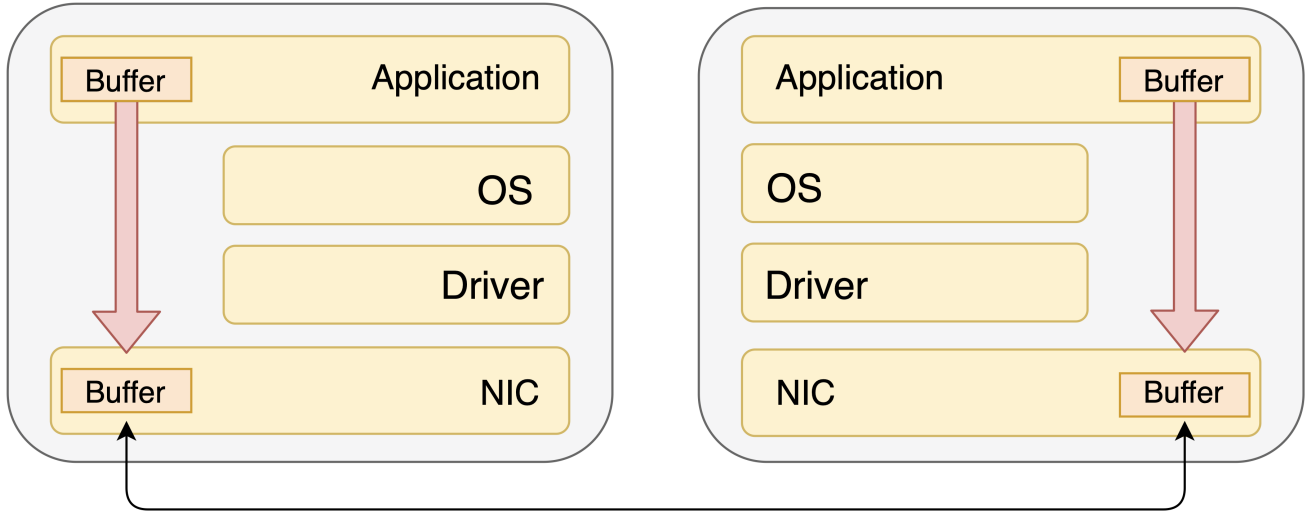


https://www.servants.co.jp/blog/technology/hno-mellanox/1742

[1] Dongarra, J., Heroux, M. A., & Luszczek, P. (2016). High-performance conjugate-gradient benchmark: A new metric for ranking high-performance computing systems. The International Journal of High Performance Computing Applications, 30(1), 3–10.

高性能な通信を支えるRDMA

- Remote Direct Memory Access (RDMA)
- 低遅延と高スループット
- カーネルへのバッファコピーを省き，リモートのCPUをバイパス
- InfiniBand, RoCE, iWARPなどで採用



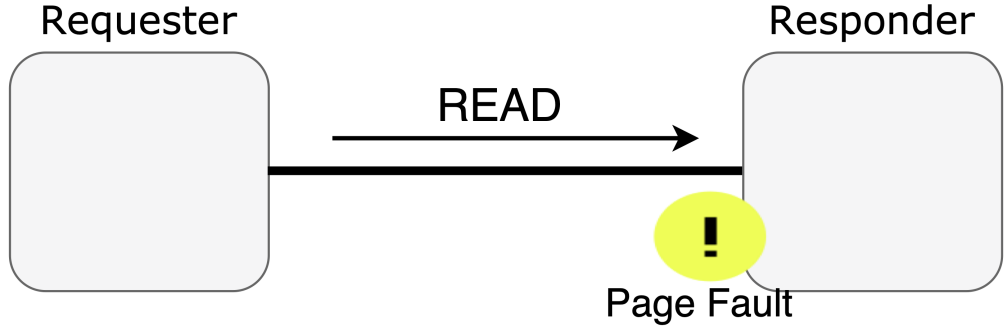
RDMAの問題点

- 通信を発行する前に，通信バッファに対してメモリレジストレーションを行う必要がある
  - バッファのスワップアウトを防ぐために物理メモリにピン留め
  - 仮想アドレスと物理アドレスの変換エントリをNICに登録
- 2つの大きな問題点
  - スワップアウトせずに計算に使用できるメモリの制限
  - バッファを使い回す際のプログラミングコストの上昇
    (cf. Pin-down cache [1])

[1] Tezuka, H., O'Carroll, F., Hori, A., & Ishikawa, Y. (1998). Pin-down cache: A virtual memory management technique for zero-copy communication. IPPS/SPDP 1998.

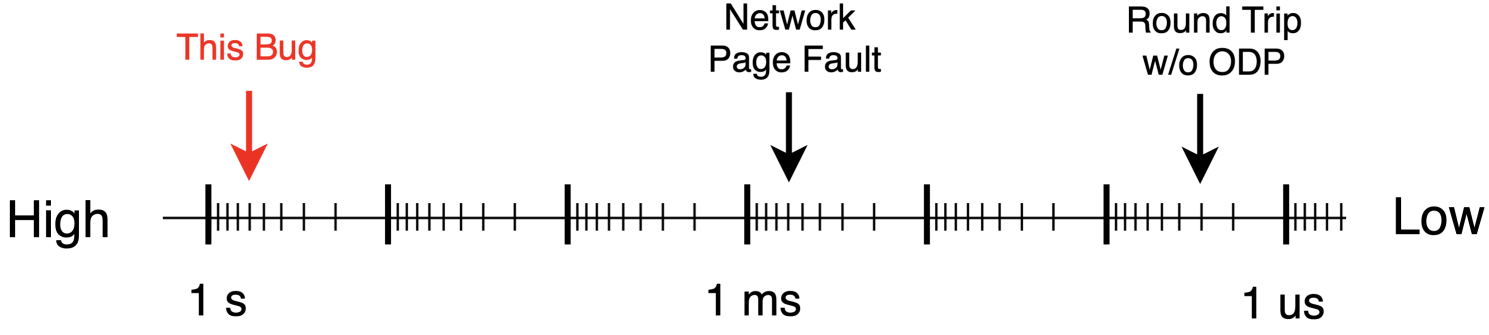
On-Demand Paging (ODP)

- Mellanoxによって最近InfiniBandに導入された前途有望な技術
  - 事前のメモリレジストレーションが不要に
- ネットワーク越しのNICのページフォルトを利用することで，通信バッファのピン留めを自動化
  - ピン留めする通信バッファのメモリ使用量の削減
  - ピン留めの手動管理が不要
- 過去の研究によると，ページフォルトに関わるオーバーヘッドは許容範囲 (数百usほど)



本研究の貢献

- ODPの重大な性能面のバグを発見
  - 驚くべきことに，単純な条件で数百msのストールが生じる
    - RDMA READを2つのみ発行という条件
  - cf. インターコネクトのレイテンシは数usほど
- ODPのバグが生じる条件とその原因を，マイクロベンチとibdumpを用いて詳細に分析
  - InfiniBandの異常に長い再送のタイムアウトが原因
- ODPのバグを回避するソフトウェアによるアプローチの提唱
  - 現時点では未実装



発表の流れ

背景

- InfiniBandと再送制御
- On-Demand Paging (ODP)

ODPの関連研究

実験

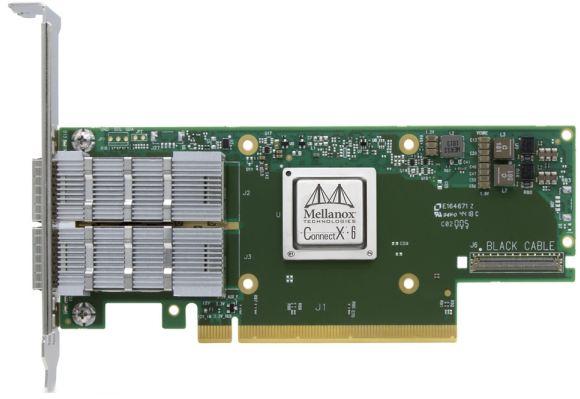
マイクロベンチによる性能バグの再現，ibdumpを用いた分析

- 通信数が2つの場合
- 通信数をさらに増やした場合

考察と結論

InfiniBand

- 主に高性能計算で用いられる，RDMAをサポートした低遅延インターコネクト
- 二種類の通信オペレーション
  - 双方向通信関数: SEND, RECEIVE
  - 片方向通信関数: READ, WRITE



https://www.infinibandta.org/tag/roce/
https://jp.mellanox.com/products/ethernet-adapters/connectx-6/

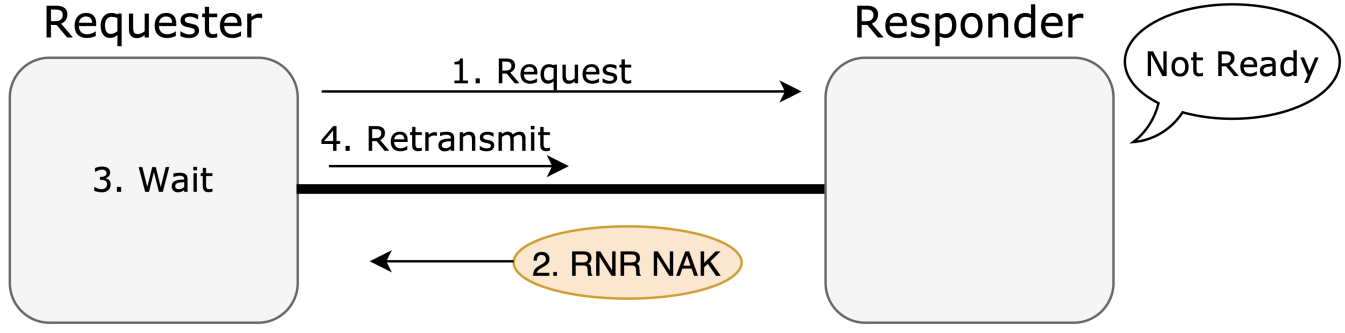
トランスポートプロトコル

- InfiniBandは4種類のトランスポートプロトコルをサポート
  - 代表的なものが，Reliable Connection (RC) と Unreliable Datagram (UD)
- RCは信頼性のあるプロトコルで，途中でエラーが生じた時には再送を行う
- 具体的に再送はどのような条件で生じるのか
  - 大きく分けて三種類が存在

再送が生じる条件 (1)

Responderの準備ができていない場合

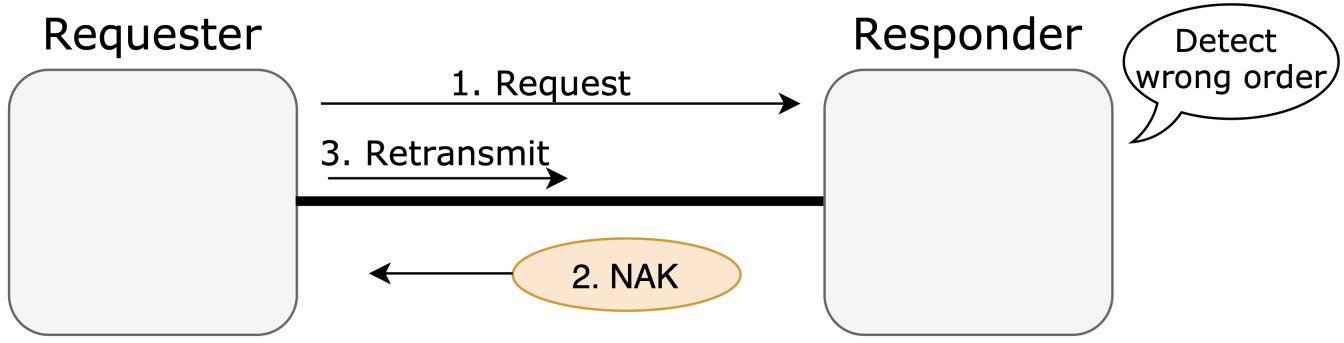
- Responderは，Receiver-Not-Ready (RNR) NAKを返し，Requesterに一定時間後 (RNR NAK delay)の再送を要求
- 例えば双方向通信の際に，RESPONDERがRECEIVEを投入していない場合



再送が生じる条件 (2)

Responderがパケットの到着順序に誤りがあるのを検出した場合

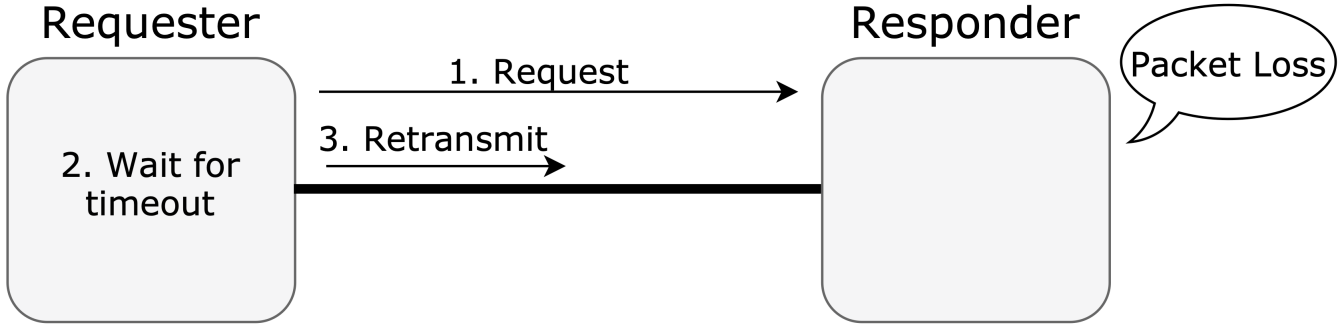
- 例えば，N+1番目のパケットがN番目のパケットより前に到着した場合には，N番目のパケットロスが消失したと考えられる
- ResponderはNAK (PSN Sequence Error) を返して，Requesterに再送を要求



再送が生じる条件 (3)

Requesterが通信を投入してから一定時間，Responseが返ってこなかった場合（これが一番重要）

- Requesterはパケットロスが生じたと判断して，再送を行う
- Responseを待つ時間（タイムアウト）は設定可能で，理論上は最短で数usに設定することができる
- ただし，下限値はIBカードのベンダーによって規定されるとのこと（伏線）



発表の流れ

背景

- InfiniBandと再送制御
- On-Demand Paging (ODP)

ODPの関連研究

実験

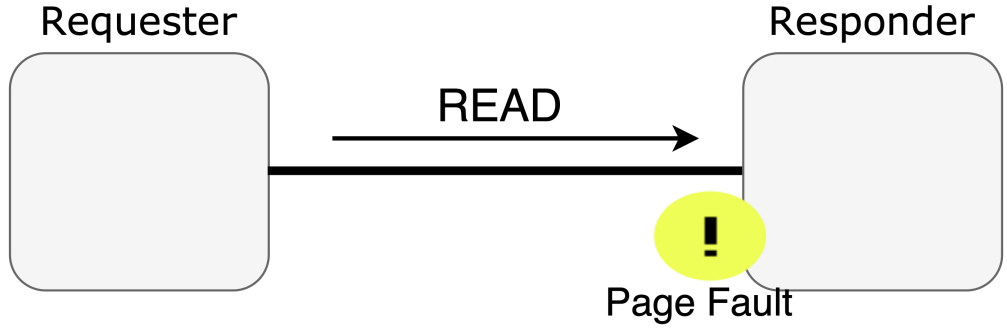
マイクロベンチによる性能バグの再現，ibdumpを用いた分析

- 通信数が2つの場合
- 通信数をさらに増やした場合

考察と結論

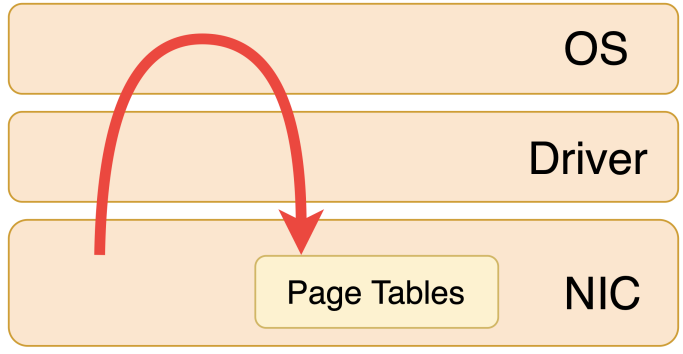
ODPの概要

- 通信バッファのピン留めを自動化して，事前のメモリレジストレーションを不要にする技術
- 通信バッファのピン留め，NICへのアドレス変換エントリの登録は，そのページが通信に必要になって初めて行う
- MVAPICH2-X, MPICH (on UCX), Open MPI (on UCX) での導入例あり



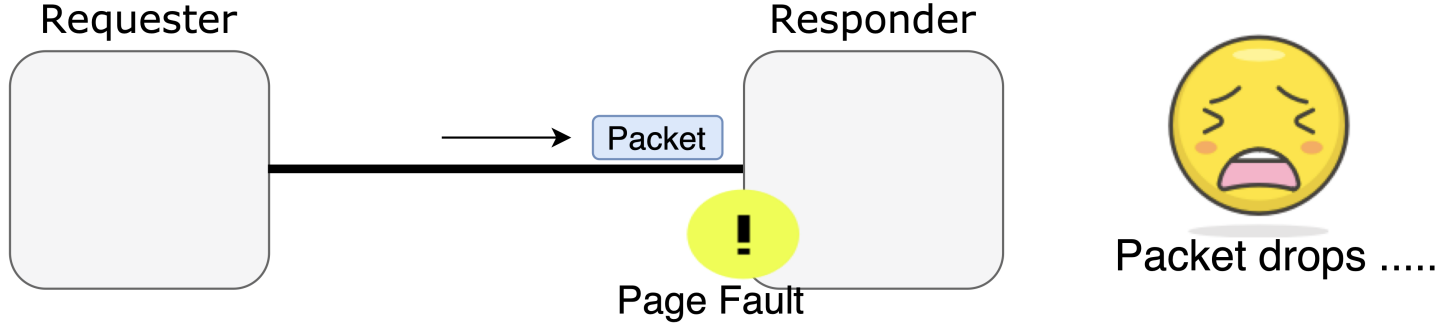
ODPの基本的な実装

- InfiniBandのドライバとNICのファームウェアで実装
- NICのページテーブルにエントリがないページにアクセス要求が来た時，次の手順で対処
  i. NICがOSにページフォールトを要求
  ii. OSは物理ページがない場合は割り当て
  iii. ドライバ経由でNICのページテーブルに，アドレス変換エントリを登録

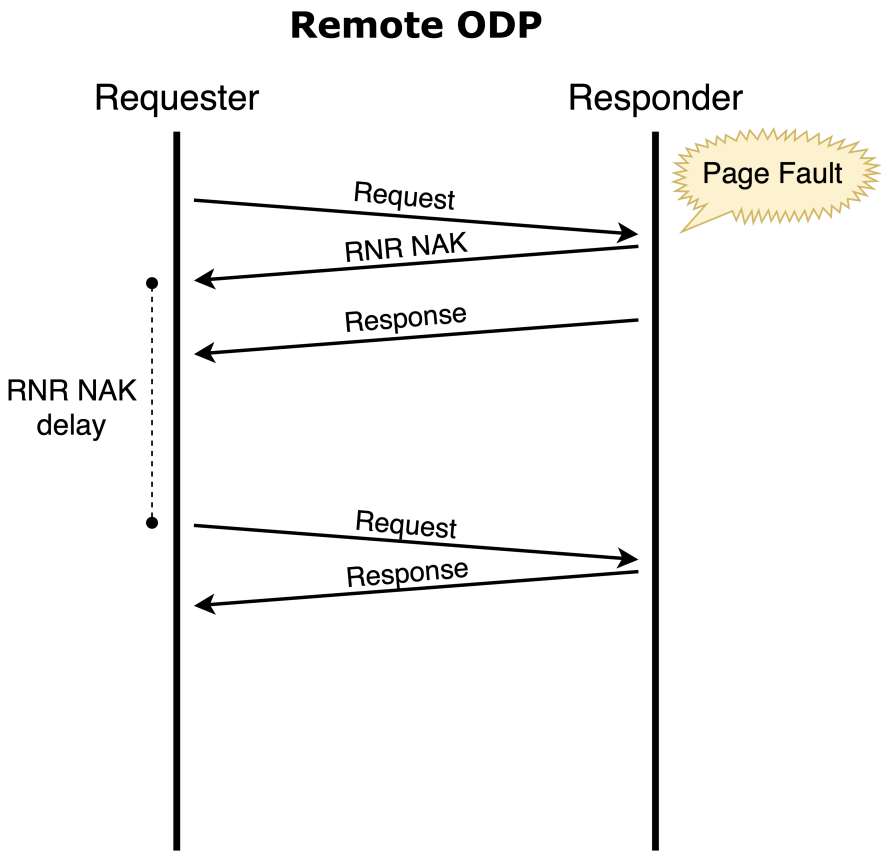


再送を用いたODP実装のサポート

- パケットの受け取り側でページフォルトが発生した場合には困難がある
  - NICのメモリサイズは限られている
  - ページフォルトが解決するまで受け取ったパケットをNICに保持することはできない
- RCの再送を利用して対処
- RDMA READを具体的にどのように対処しているのかibdump（パケットキャプチャツール）で観察



RDMA READの挙動



- Responder側のみODP (Requester側は時間の都合上割愛)
- Responderがページフォルトを引き起こすパケットを受け取ると，RNR NAKを返して再送を要求

発表の流れ

背景

- InfiniBandと再送制御
- On-Demand Paging (ODP)

ODPの関連研究

実験

マイクロベンチによる性能バグの再現，ibdumpを用いた分析

- 通信数が2つの場合
- 通信数をさらに増やした場合

考察と結論

ODPの関連研究

ODPの実装と性能分析に取り組んだ研究を紹介

- Lesokhinらは，ODPそのものを提唱 [1]
  - ページフォルトのオーバーヘッドは数百usであり，ODPのコストは許容可能
- Liらは，ODPをMPIに取り入れて評価 [2,3]
  - ODPを使用しない時と比較して，少ないメモリ量で同程度の性能
  - RNR NAKで指定する待ち時間 (RNR NAK delay) を短くすることで，ページフォルトの時間を軽減可能

[1] Lesokhin, I., Eran, H., Raindel, S., Shapiro, G., Grimberg, S., Liss, L., ... Tsafrir, D. (2017). Page Fault Support for Network Controllers. ASPLOS'17.

[2] Li, M., Hamidouche, K., Lu, X., Subramoni, H., Zhang, J., & Panda, D. K. (2016). Designing MPI Library with On-Demand Paging (ODP) of InfiniBand: Challenges and Benefits. SC'16.

[3] Li, M., Lu, X., Subramoni, H., & Panda, D. K. (2017). Designing Registration Caching Free High-Performance MPI Library with Implicit On-Demand Paging (ODP) of InfiniBand. HiPC'17.

発表の流れ

背景

- InfiniBandと再送制御
- On-Demand Paging (ODP)

ODPの関連研究

実験

マイクロベンチによる性能バグの再現，ibdumpを用いた分析

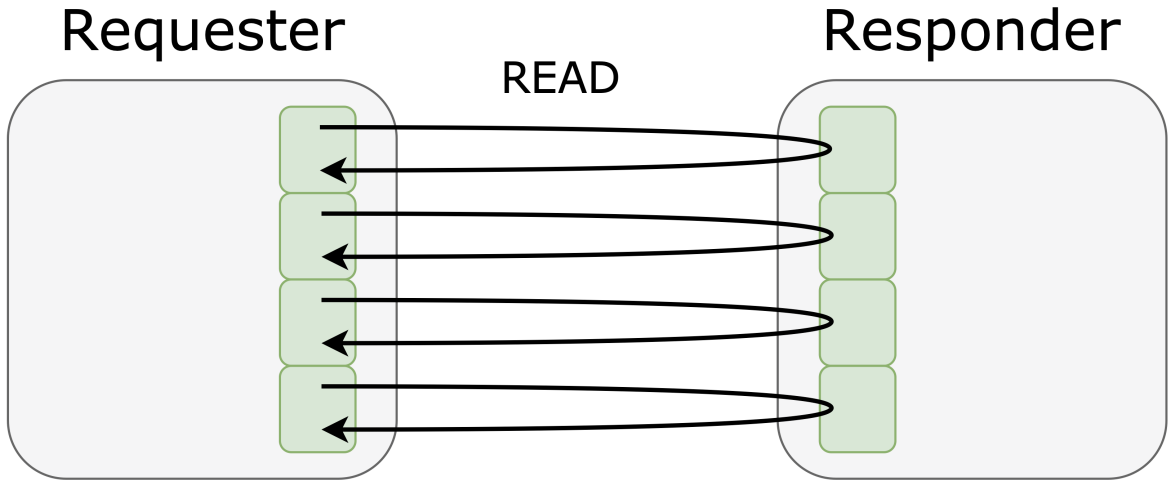
- 通信数が2つの場合
- 通信数をさらに増やした場合

考察と結論

実験環境

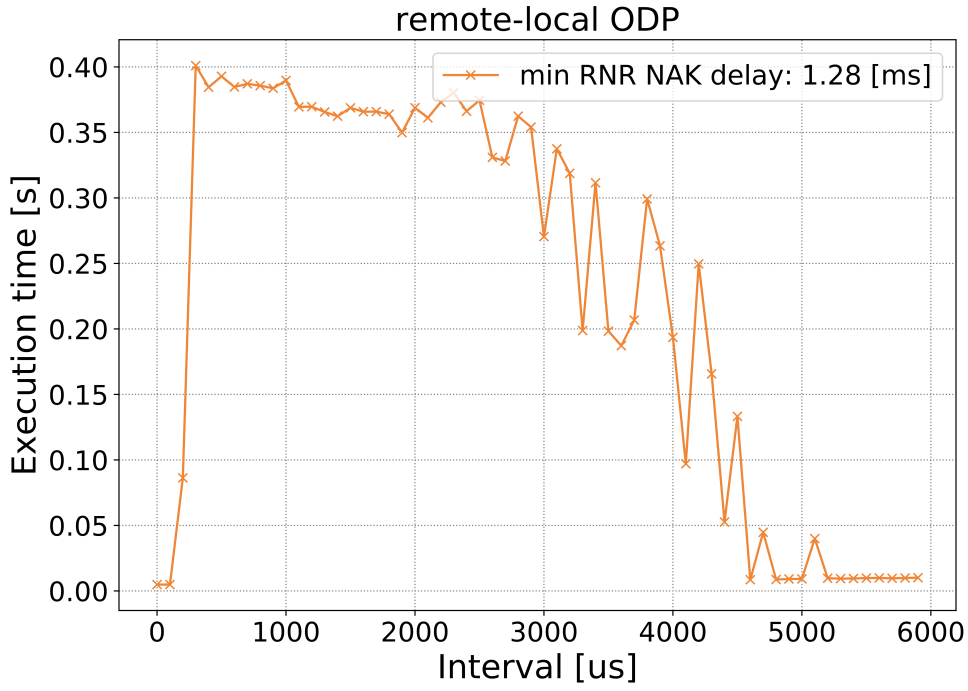
- ２つのマシンをInfiniBandで直接接続
- Xeon Phi CPU 7250 (1.40 GHz, 272スレッド)
- PC4-19200 196GB, MCDRAM 16GB
- Mellanox MCX456A-FCAT ConnectX-4 VPI adapter
- 再送のタイムアウトの設定は最小値を使用

マイクロベンチマーク



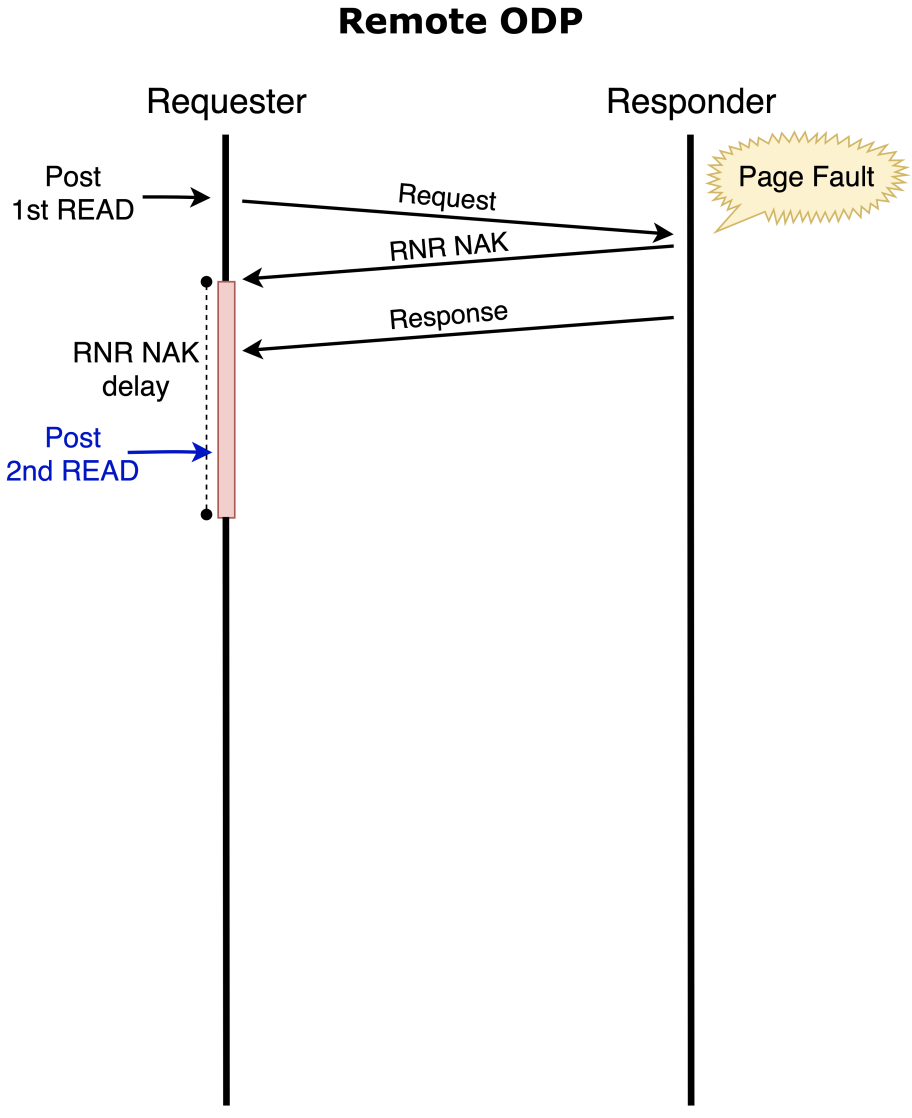
- 片方のマシンがもう一方にRDMA READを複数発行
- 通信の発行する間隔 (Interval)と，通信の発行数を変更
- 1つの通信のサイズは100 bytes

通信数が2つの場合



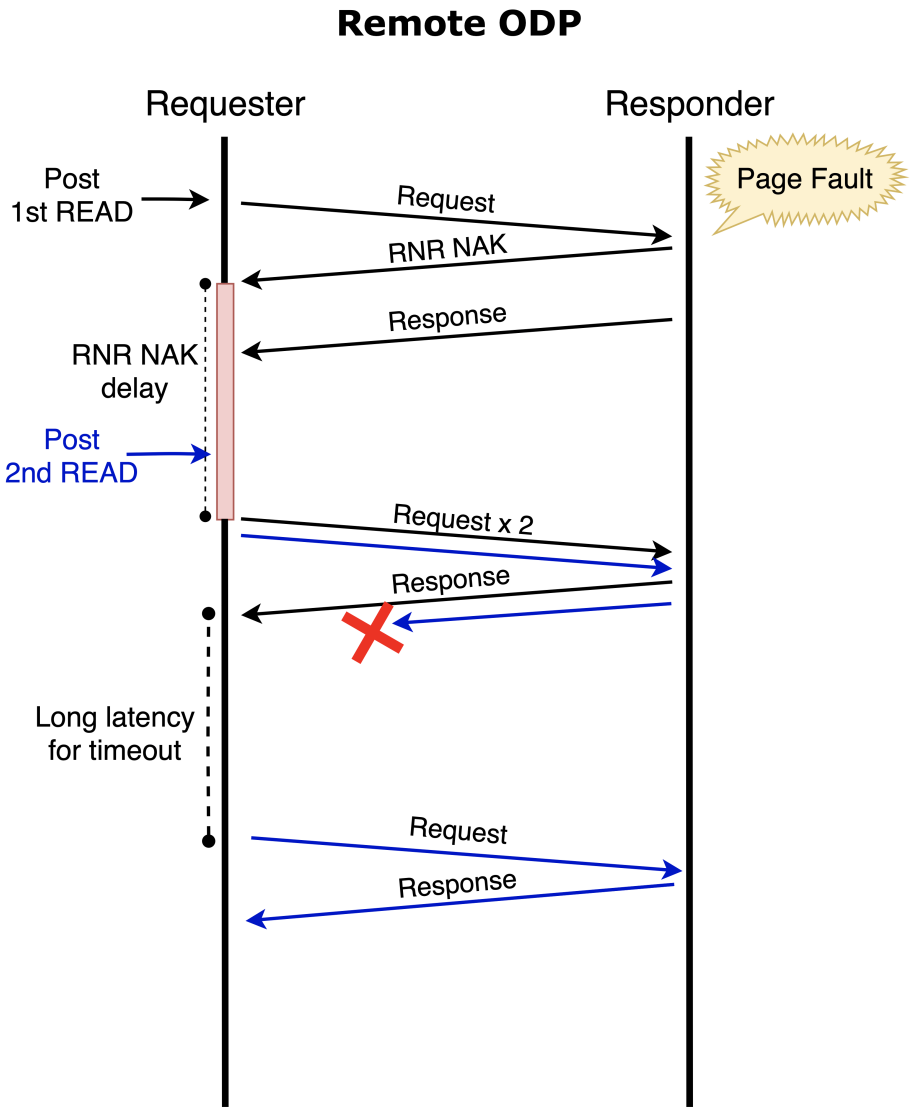
- Requester, Responder両方についてODPを設定
- Intervalを変更して，10回実行した時の実行時間の平均をプロット
- Intervalが500usから4500usの時には，実行時間が数百msほどと非常に長い

遅延が大きい時のRemote ODPの挙動



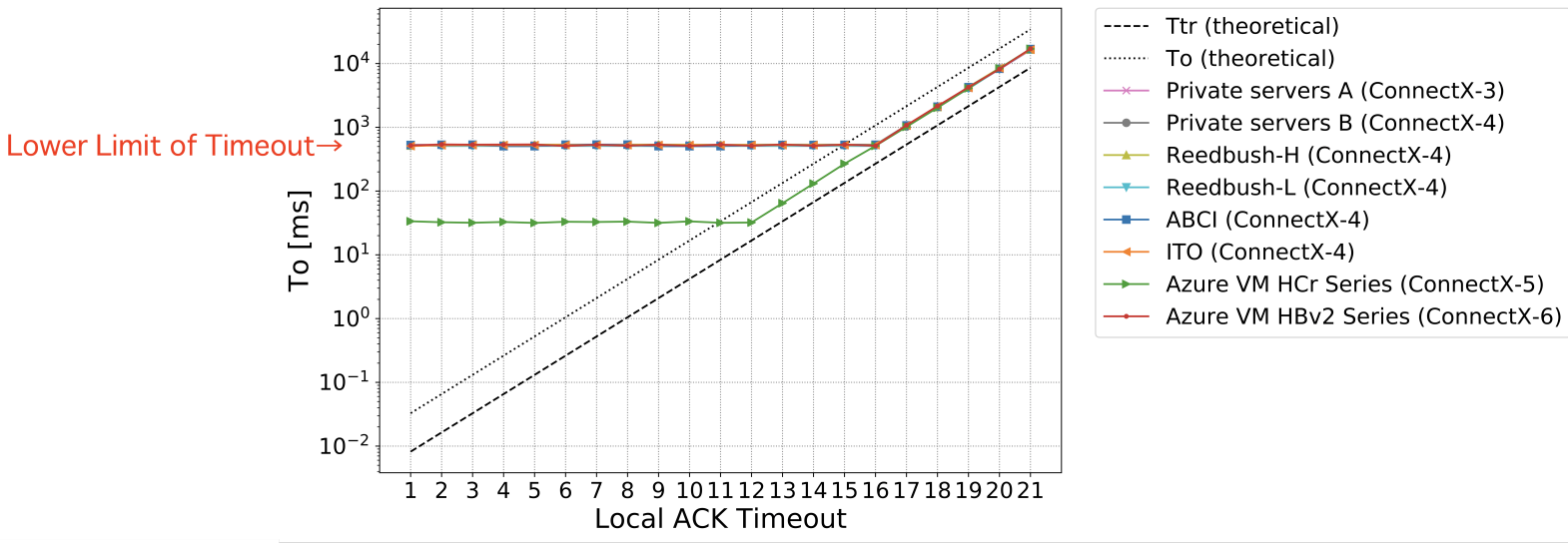
- ibdumpで分析
- Local ODPは割愛（同様）
-
-

遅延が大きい時のRemote ODPの挙動



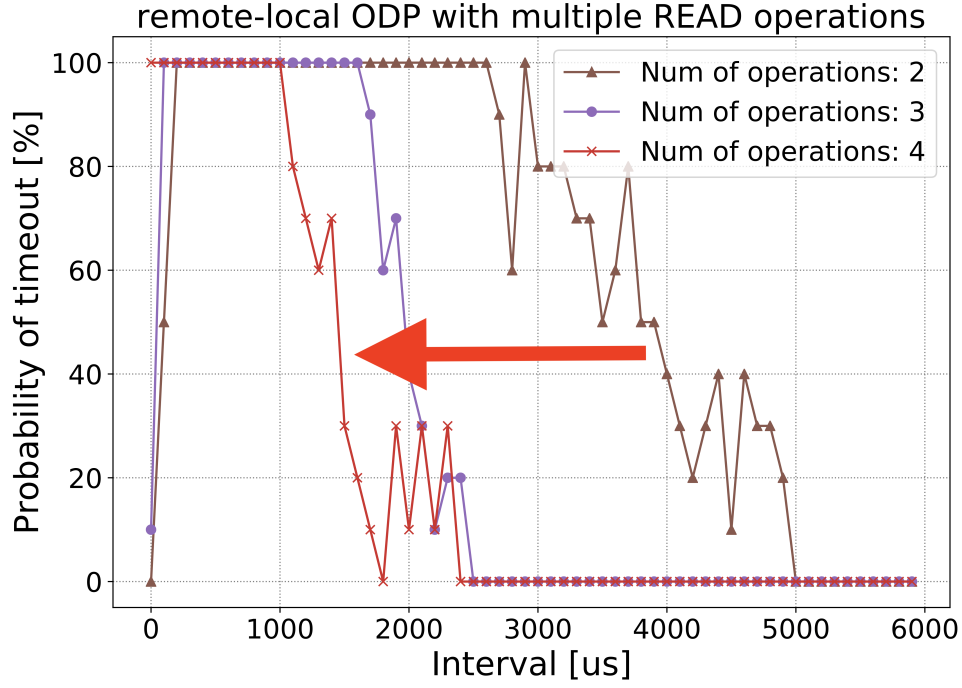
- ibdumpで分析
- Local ODPは割愛（同様）
- RNR NAKの待ち時間 (RNR NAK delay) の間に２つ目のパケットが来ると，その後にパケロス
- タイムアウトまで待って再送，しかしこのタイムアウトが異常に長い（なぜ？）

タイムアウトは最小に設定したはずでは？



- タイムアウトの下限値を実際に測定
- 面白いことに，タイムアウトの下限値は最新の機種を含めたほとんどの機種で500msほどと，非常に大きな値だった（伏線回収）
- これが性能バグの元凶

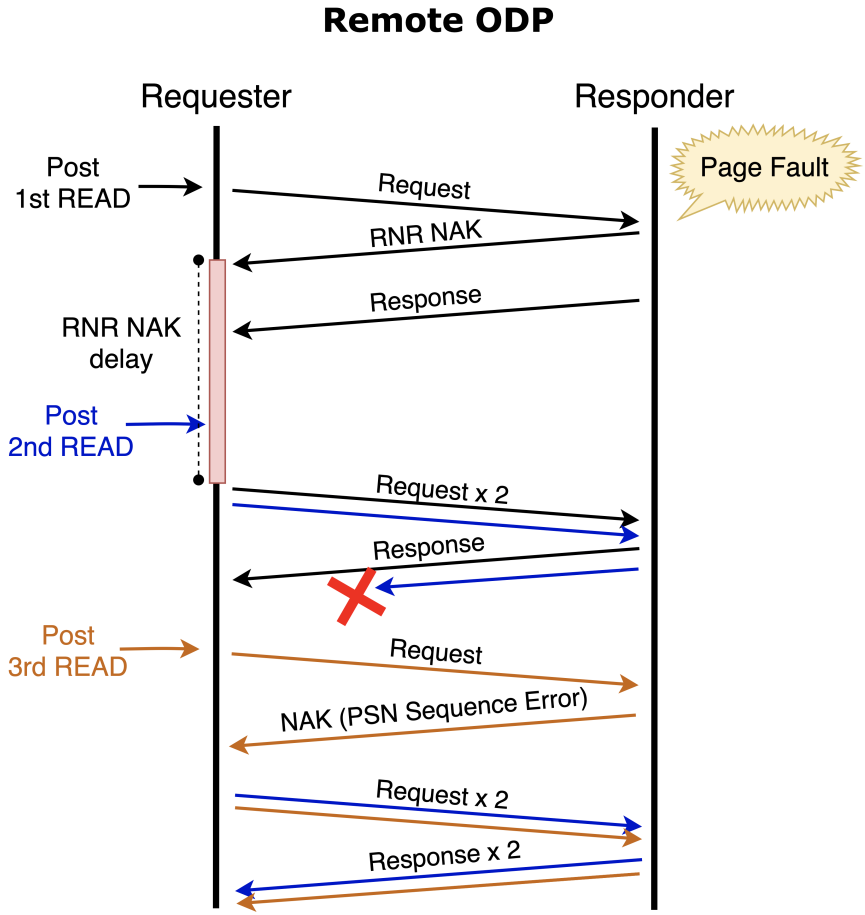
通信数をさらに増やした場合



- 通信数を2から4まで変更して同様に測定
- 不思議なことに，タイムアウトが生じる間隔（Interval）のレンジが小さくなる（生じにくくなる）

通信数が3つの場合の分析

3つ目のパケットがタイムアウトの待ち時間に発行された場合の挙動



- PSN Sequence Errorが検出され、即座に再送が行われる
- タイムアウトによるストールはなんと消失

その他の条件での実験

- SEND, WRITEなどの，READ以外のオペレーションでは生じない
- 通信のサイズには関係なく生じる
    - 1つ目の通信でページフォルトが生じれば，それ以降の通信はどんなものでも構わない
- Reedbush-H/L, ABCI, ITOを含む，様々なシステムで生じる

発表の流れ

背景

- InfiniBandと再送制御
- On-Demand Paging (ODP)

ODPの関連研究

実験

マイクロベンチによる性能バグの再現，ibdumpを用いた分析

- 通信数が2つの場合
- 通信数をさらに増やした場合

考察と結論

考察

- ODPでタイムアウトは日常的に生じていてもおかしくない
- この性能バグがODPの実装の制約から起因する問題なのかは不明
  - READでしか生じない理由は不明
  - ドライバのコードを見るも，ドキュメント不足で解読できず
  - ここを深く追求するのは正直厳しい
- そもそもIBのタイムアウトはなぜ数百msより小さくできないのか
  - タイムアウトについて言及している論文は発見できず
  - InfiniBandはリンクレベルでのロスレスのフロー制御を導入していて、通常パケロスが問題にならないためか

考えられる解決策

- 3つ以上通信を発行した場合の結果に注目
- パケロスを検知した場合に新しく通信を投入して，意図的にNAKを返させるような，ソフトウェアレイヤの開発
- 既存にすでに動いているスパコンシステムでも導入可能な手法

結論

- ODPには数百msほどの間ストールする性能バグが存在し，パケットロスと，それに続く異常に長いInfiniBandの再送のタイムアウトが原因である
- このパケットロスは，RDMA READによって生じたODP機構の待ち時間の間に，別の通信が投入されることによって生じる
- 薄いソフトウェアレイヤの導入で解決できる見込みあり

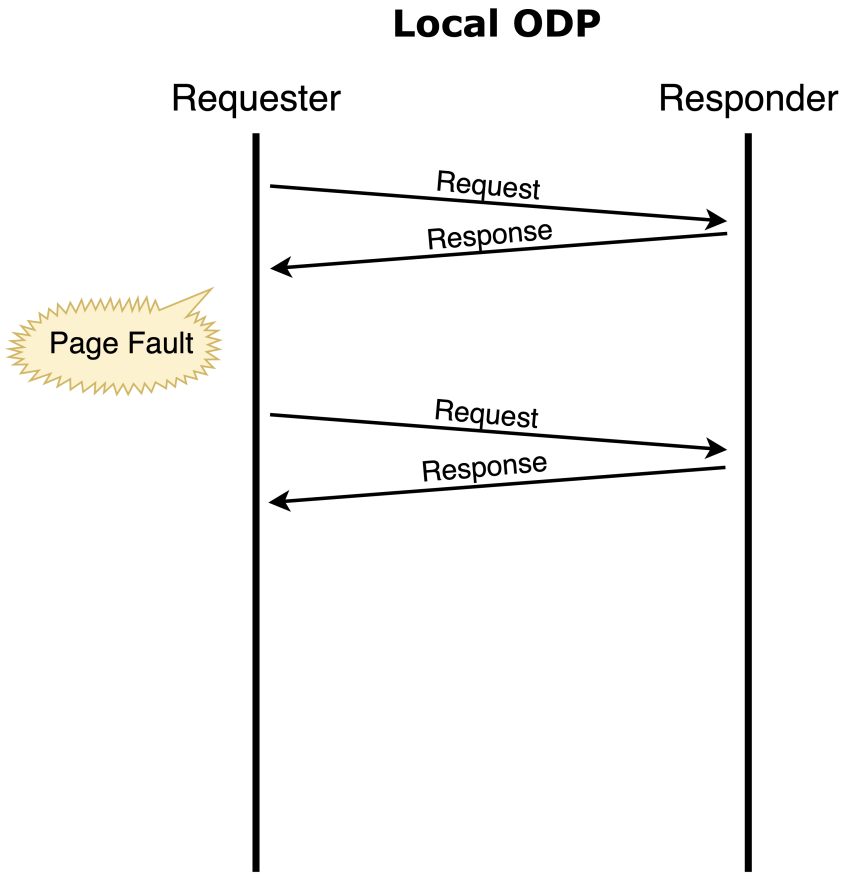
今後の方針

- ソフトウェアでの解決方法が本当に機能するのか検証
  - ハードウェアの同時通信発行数の制限がネックになると予想
  - ソフトウェアのオーバーヘッドの測定が必要
- 性能バグが生じる，適切なアプリの発見
  - RDMA READを用いるアプリであることは不可欠
  - 遅延が性能と直結し，細かい粒度での頻繁な同期が求められるものが良いのではないか
    - 3つ以上の通信が発行された場合は出現確率が下がるため
  - 現状，ArgoDSM (on MPI RMA on UCX) [1] では再現
- 研究としてどこに売り出すべきか検討
  - 少なくとも性能面で著しい影響がある問題ではある
  - IBの異常に長いタイムアウトに対し問題提起するのは有意義

[1] https://github.com/etascale/argodsm

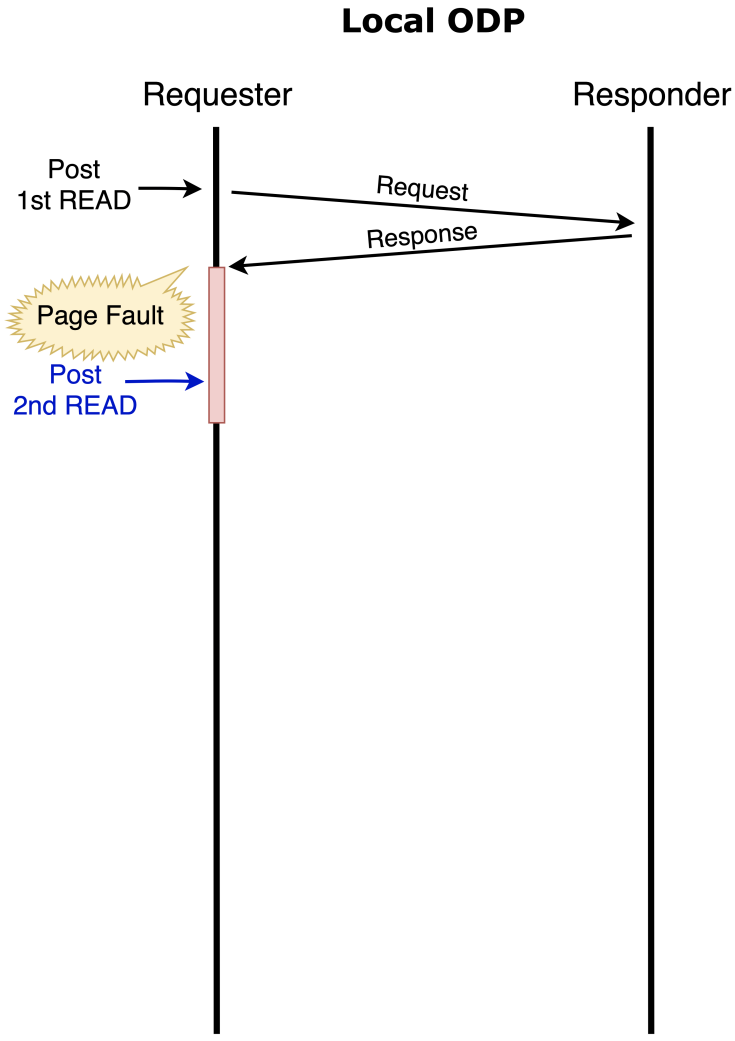
予備スライド

Local ODPの挙動



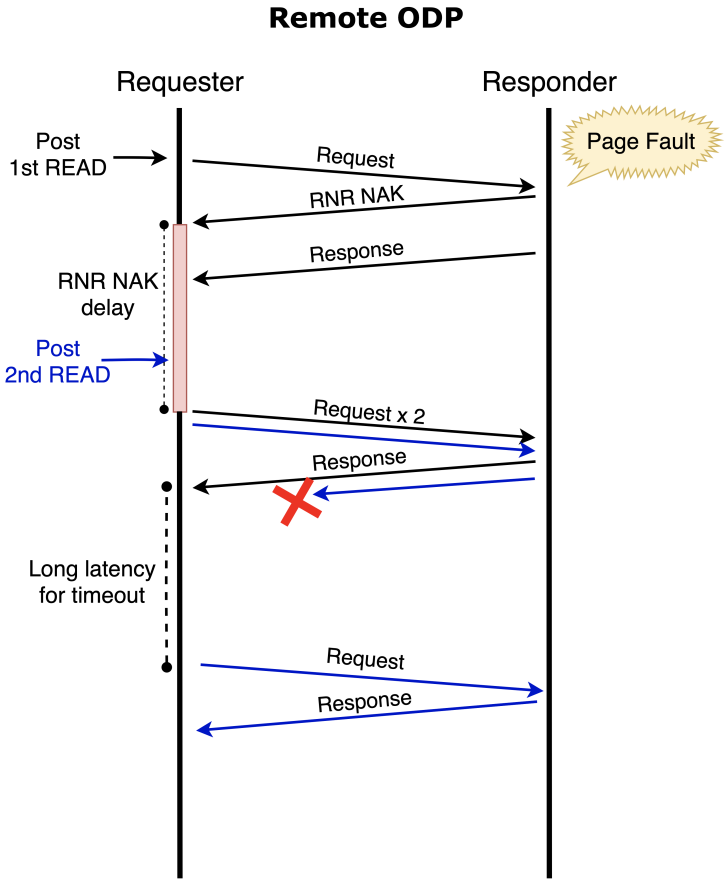
- Requester側のみODP
- パケットがResponderから返ってきた直後にページフォルトが生じ，それが解決された後にRequester自らが再送

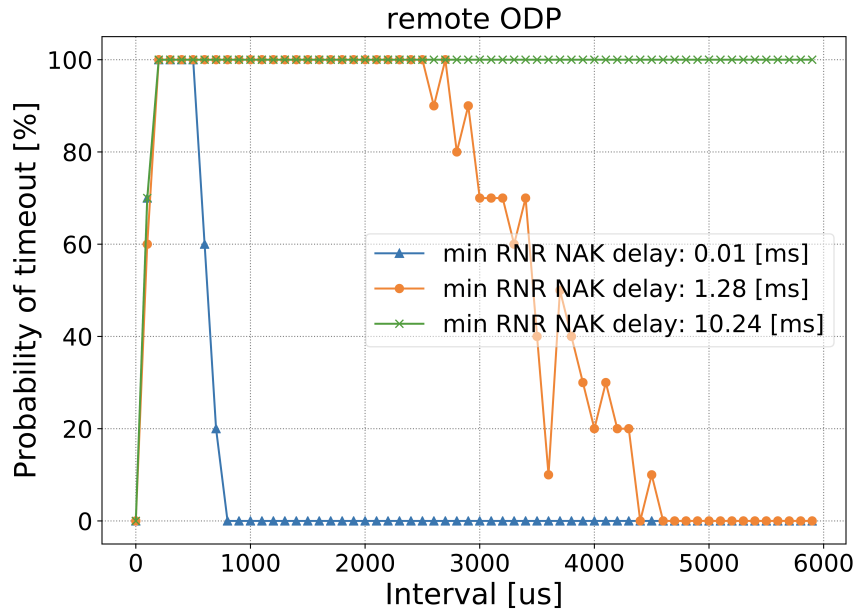
遅延が大きい時のLocal ODPの挙動



- ローカルのページフォルトを解決している時に２つ目のパケットが来ると，その後にパケロスする

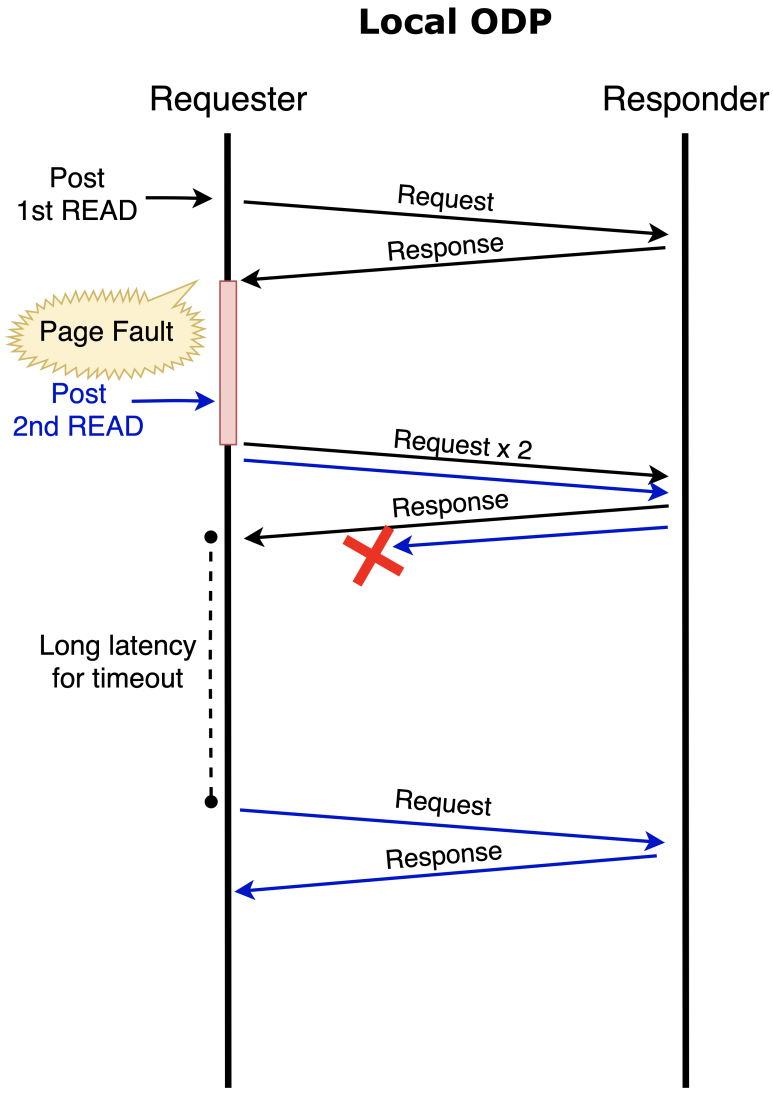
Remote ODP（RNR NAK delayを変化）





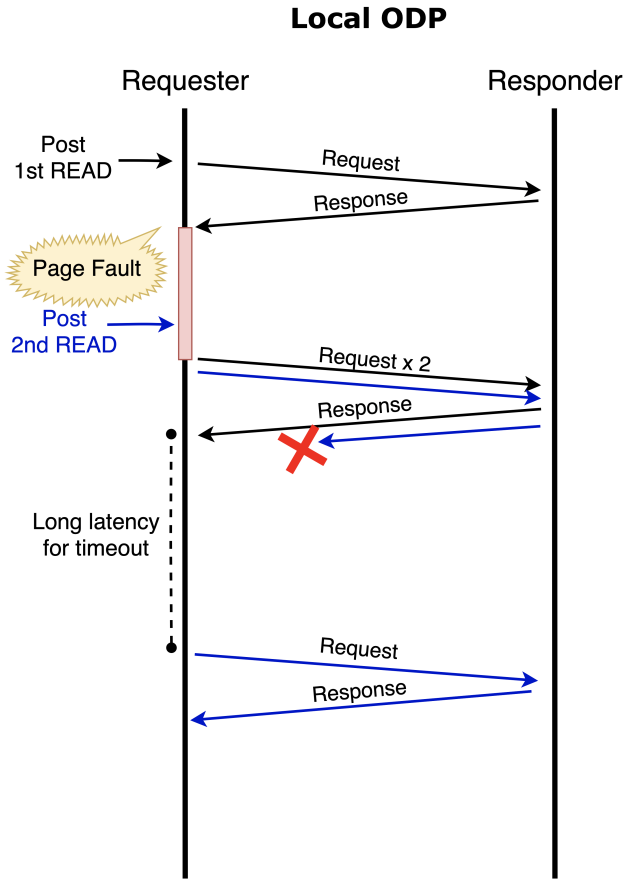
- RNR NAK delayを変更すると，それに応じてタイムアウトが生じる範囲も変化
- 1000usほどまでに性能バグの生じるIntervalの上限を縮めることが可能

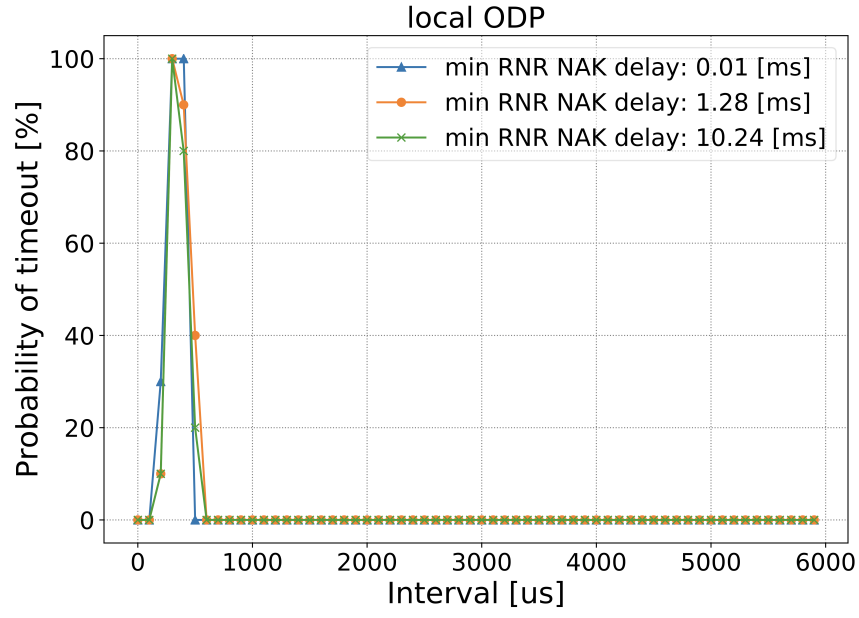
遅延が大きい時のLocal ODPの挙動



- ローカルのページフォルトを解決している時に２つ目のパケットが来ると，その後にパケロスする

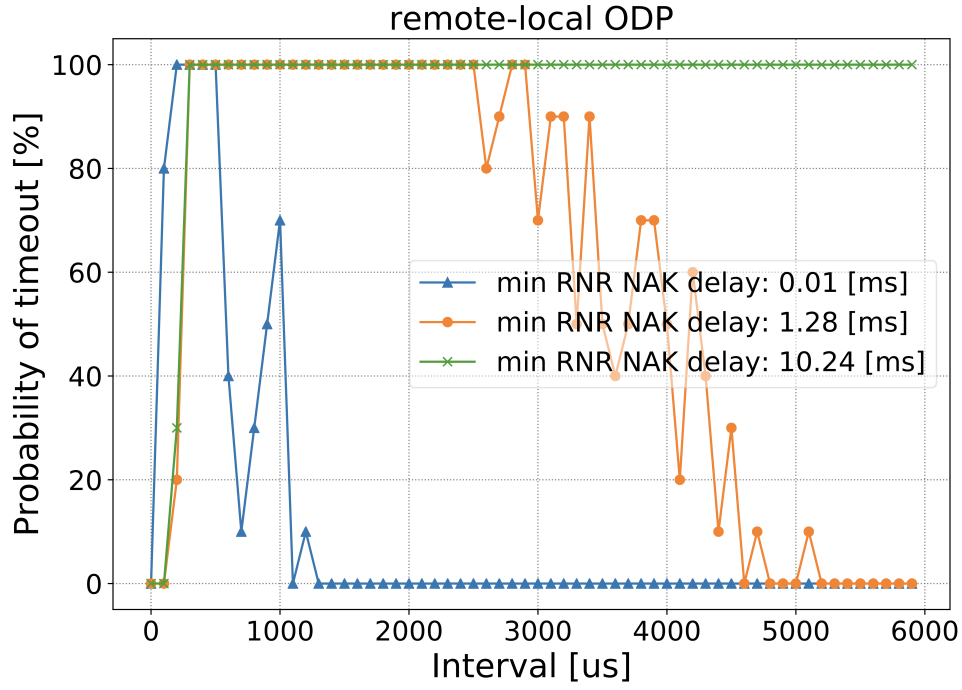
Local ODP（RNR NAK delayを変化）





- Requester側のODPのみだと，性能バグが生じるIntervalの範囲は100-500us程度
- RNR NAK delayは特に関係がない

Remote-local ODP（RNR NAK delayを変化）



- Requester側とResponder側両方にODPを適用すると，両方の性能バグが生じるレンジを合わせたレンジで生じる