Pitfalls of On-Demand Paging of InfiniBand

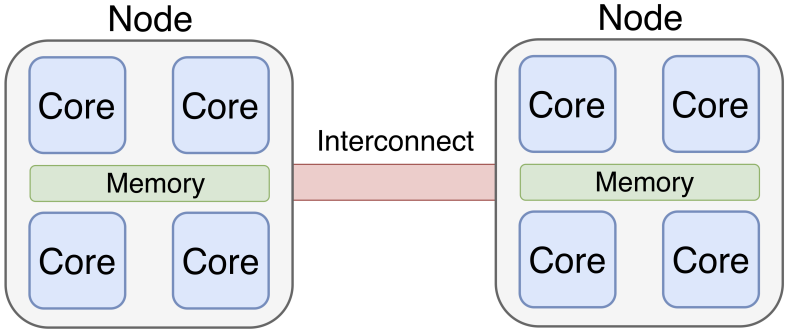
ISPASS 2021, March 30

Takuya Fukuoka, Shigeyuki Sato, and Kenjiro Taura

The University of Tokyo

The demand for high-performance interconnects

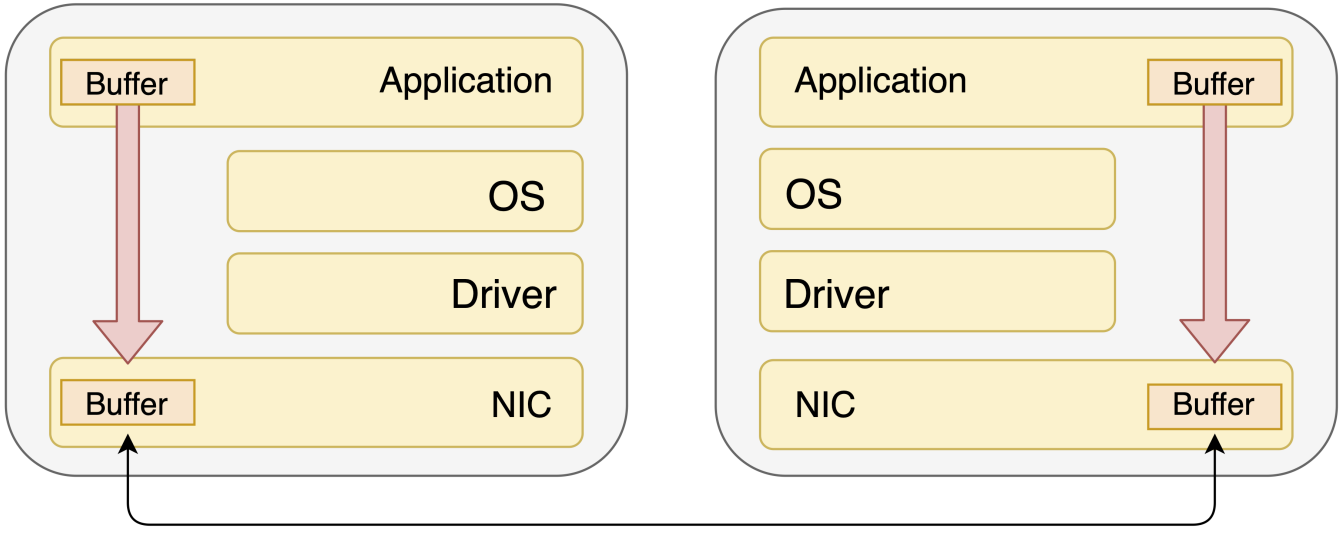
- Computing using distributed environment is popular
  - AI, BigData, scientific computation and so on
  - Platform: datacenters and supercomputers
- The overall performance is affected by interconnects
  - InfiniBand, high-spped Ethernet, Intel Omni-Path



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

Remote Direct Memory Access (RDMA)

- Technology of interconnects
  - Low latency and high throughput
  - Avoid the buffer copies and bypass the remote CPU
- Need to register memory of the communication buffers before issuing communications
  - Called memory registration
- Introduced in InfiniBand, RoCE, and iWarp



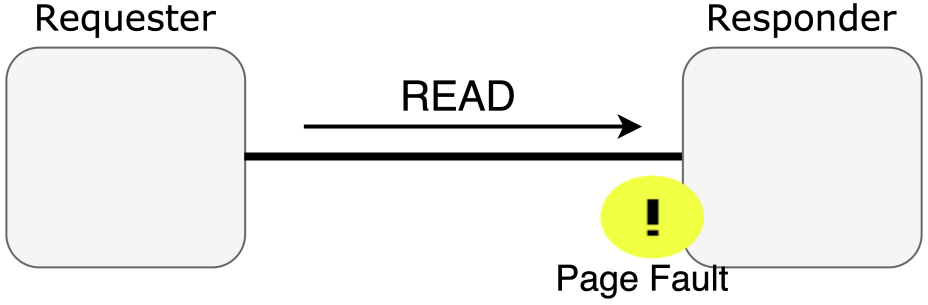
Problems of memory registration

- The memory regions which are being registered cannot be swapped out
- Two major problems
  - Less memory available for the computation
  - High programming cost to manage communication buffers (e.g. 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)

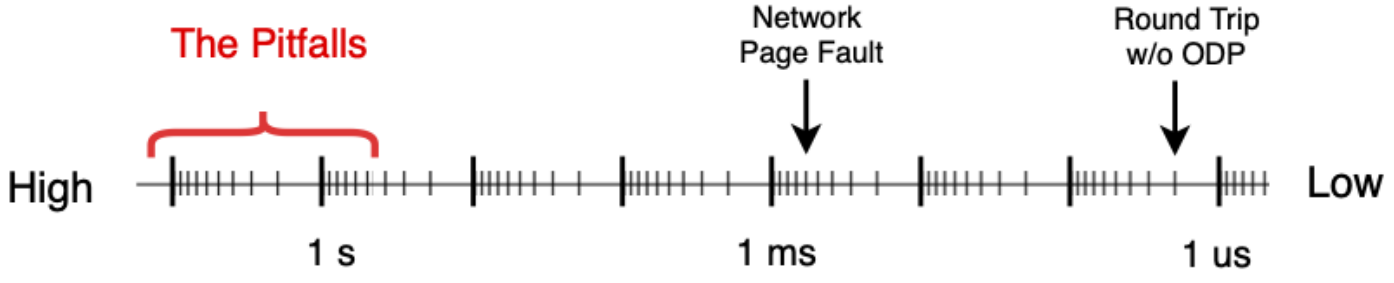
- Emerging technology recently introduced to InfiniBand by NVIDIA
  - No need to register memory beforehand
- Under ODP, memory registration is triggered by page faults on NICs only when needed
  - Only regions that are actually used can be registered without manual memory management
- The previous work reported the overhead of page fault is acceptable (around several hundred microseconds) [1]



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

Contributions of this work

- Reverse-engineer the behavior of ODP by observing packets
  - ODP has not been researched so much
- Find two critical performance pitfalls of ODP
  - Surprisingly, stall of several hundred milliseconds to several seconds appear in simple conditions
    - cf. the latency of interconnects is basically several microseconds
  - Identify the situations and causes using microbenchmarks
- Confirm these pitfalls can appear in real systems (see paper)
  - Target systems: ArgoDSM and SparkUCX



Outline

Background

- InfiniBand
- On-Demang Paging (ODP)

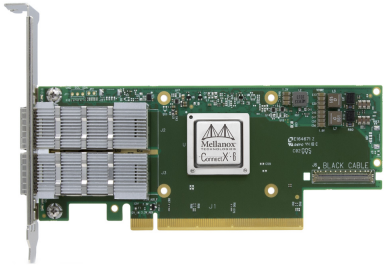
Experiments

- The first pitfall: packet damming
- The second pitfall: packet flood

Summary

InfiniBand

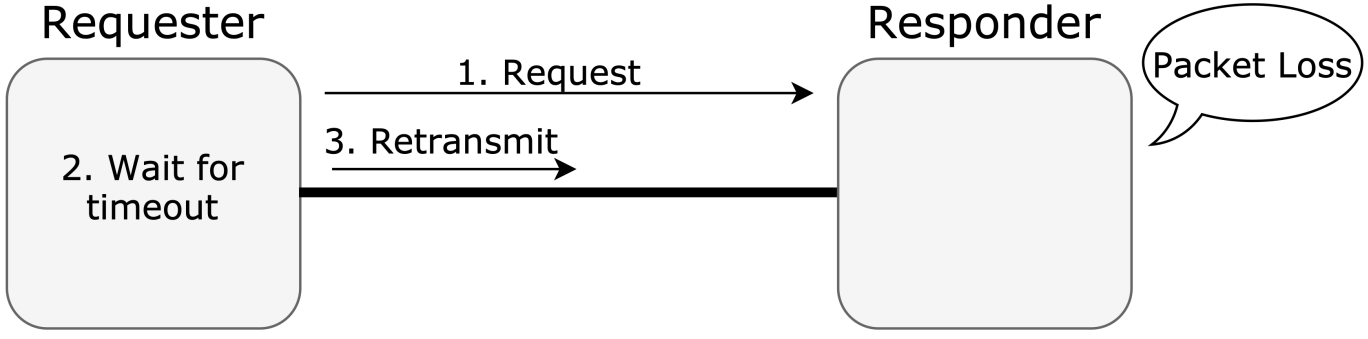
- An interconnect with ultra-low latency mainly used for High-performance Computing
  - Support RDMA
- Two kinds of communication operations
  - Two-sided: SEND, RECEIVE
  - One-sided: READ, WRITE
- Each operation is posted into a QP (Queue Pair)
  - A QP is a communication resource



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

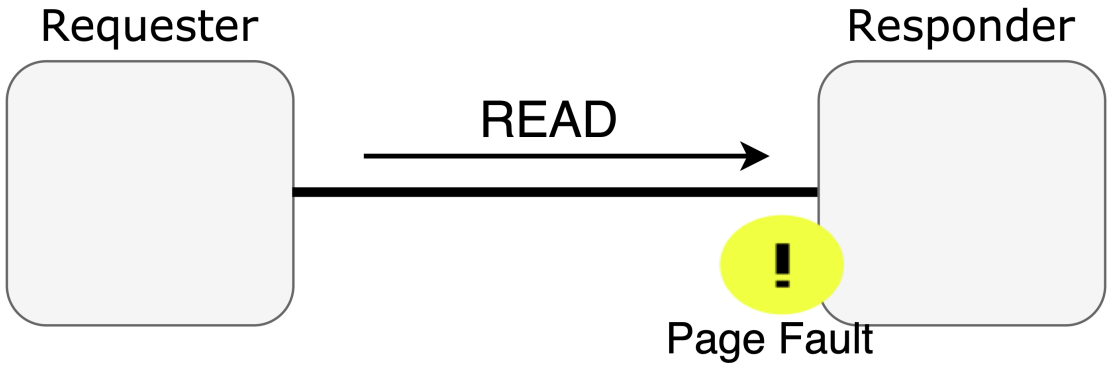
Transport layer of InfiniBand

- InfiniBand supports four kinds of transport protocols
  - Reliable Connection (RC) and Unreliable Datagram (UD) are famous
- RC is a reliable protocol and supports retransmission when an error occurs
  - ex. some packets are lost and the timeout occurs



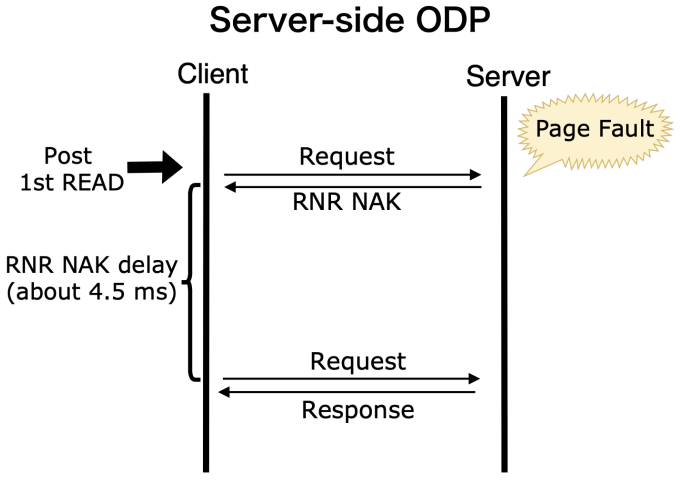
On-Demand Paging (ODP) of InfiniBand

- An extension of InfiniBand to eliminate the need for memory registration beforehand
  - On-demand memory registration by hardware only when needed
- Introduced in some MPI libraries



Implementation of ODP

- Being implemented on the driver and firmware in NICs
- The details about how ODP works is unclear
  - Only the fact that ODP utilizes retransmission of RC is public
  - The behavior of one-sided operations is especially unclear
- We investigated the behavior of ODP by reverse-engineering



Two performance pitfalls of ODP

Through this in-depth investigation of ODP, we found two performance pitfalls

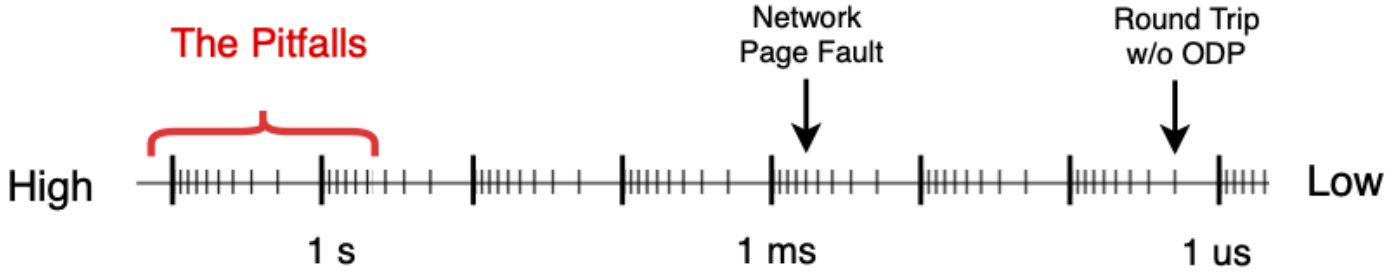
- Packet damming
- packet flood

Experimental environment

- Connected two machines with InfiniBand
  - A server and a client
- Xeon Phi CPU 7250 (1.40 GHz, 272 threads)
- PC4-19200 196GB, MCDRAM 16GB
- MCX456A-FCAT ConnectX-4 VPI adapter
- Set the smallest value to the timeout

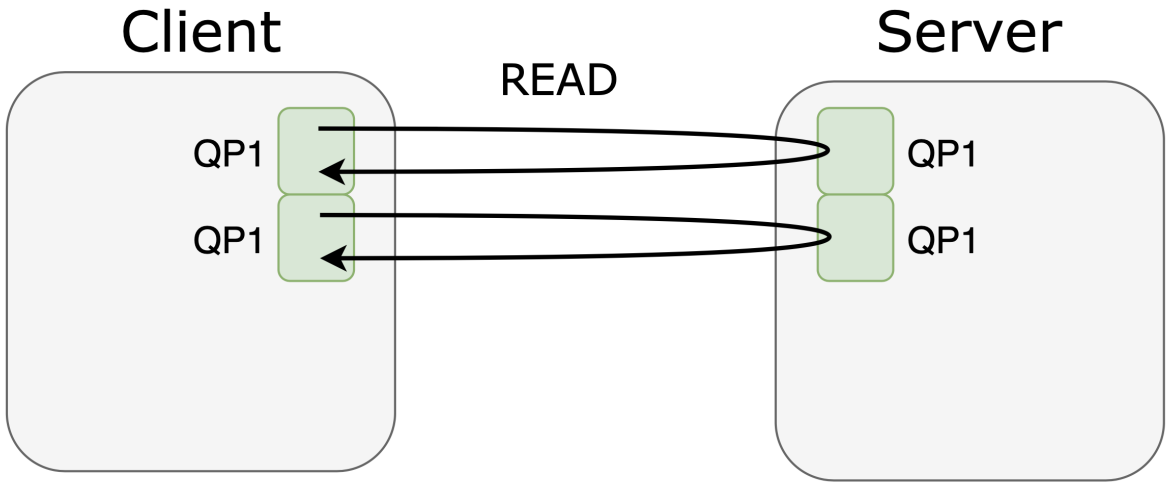
The first pitfall: packet damming

- Situation
  - Basically issue two READs with a certain interval
- Characteristics and effect
  - Communication packets get stuck (dammed) for several hundred milliseconds
- Cause
  - Packet loss and the subsequent super-long timeout



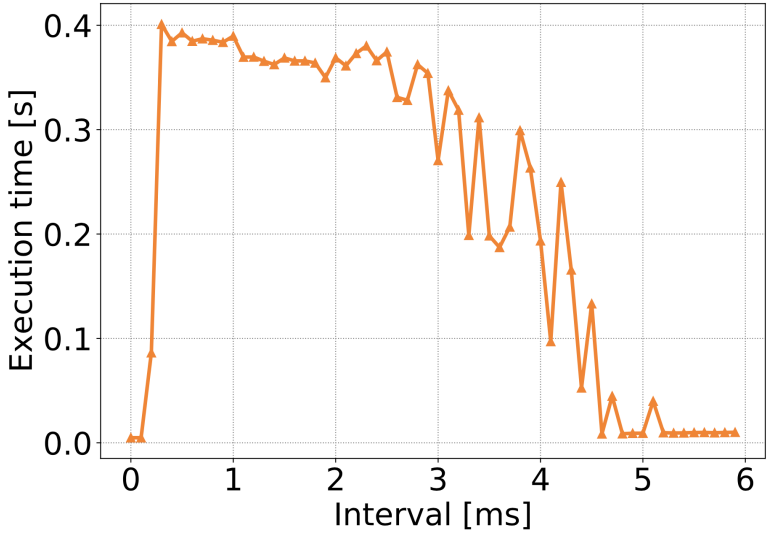
- By the way, we found it by analyzing another distributed system
  - Took several months to identify the root cause is ODP

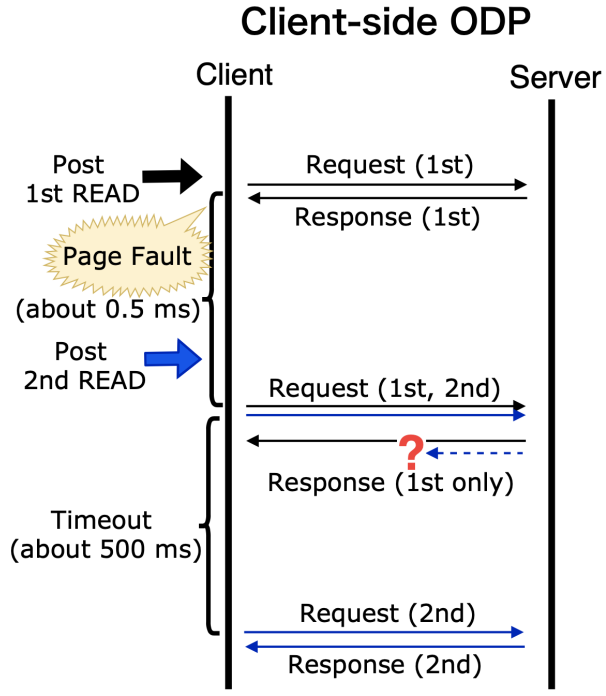
Microbench for packet damming



- One machine issues two READs to the other machine
  - Very simple situation
- Change the interval between two communications
- The message size is 100 bytes, use a single QP, and apply ODP to both sides

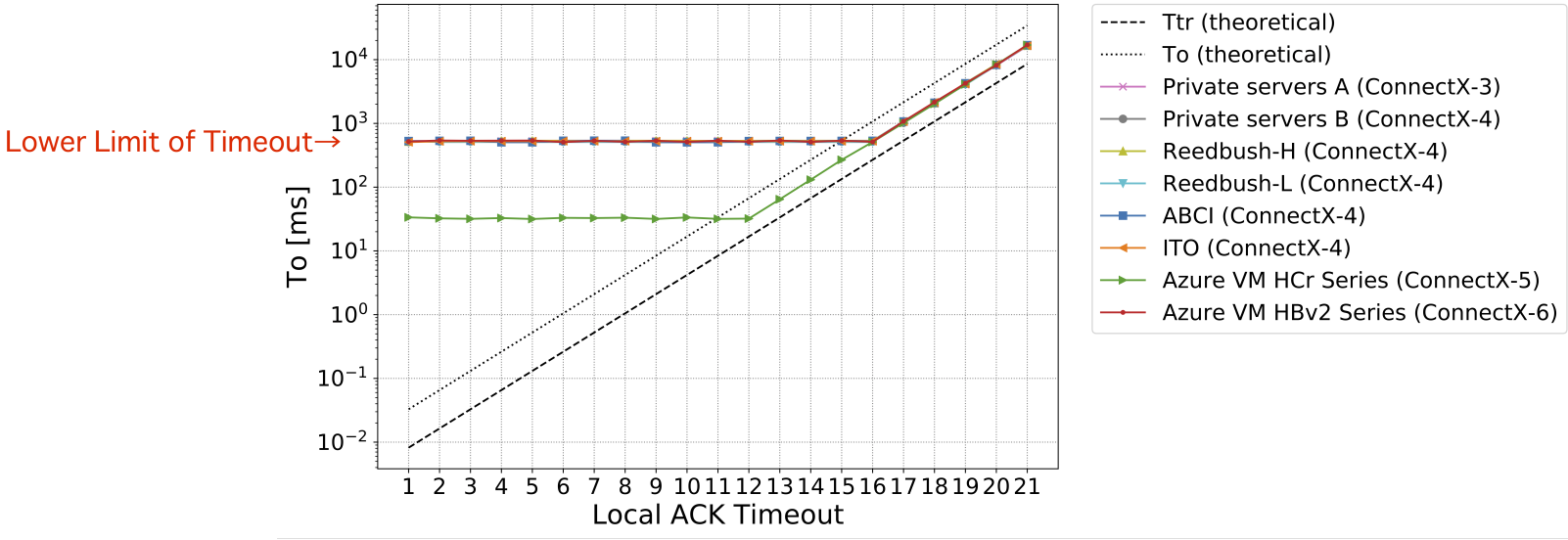
Execution time of the microbench





- The execution time is around several hundred milliseconds with the intervals of 500us to 4500us
  - This is unexpectedly too long
- Our detailed investigation shows that the packet loss and subsequent timeout happens

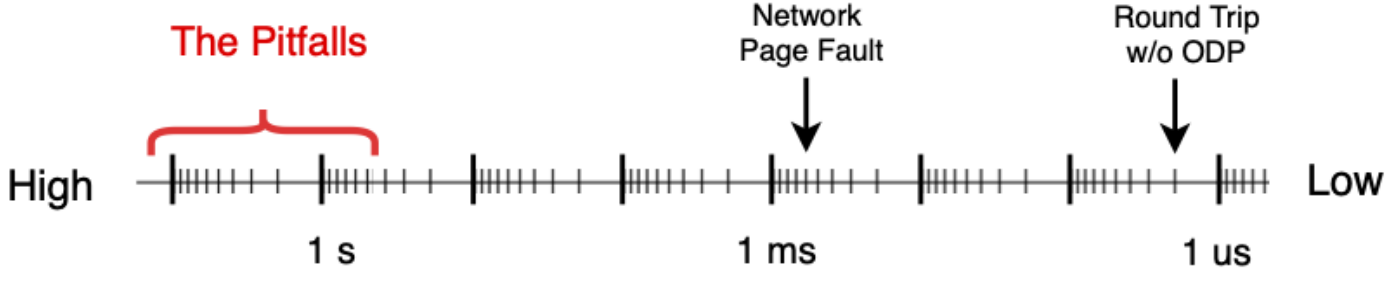
Shouldn't the timeout have been set to the minimum?



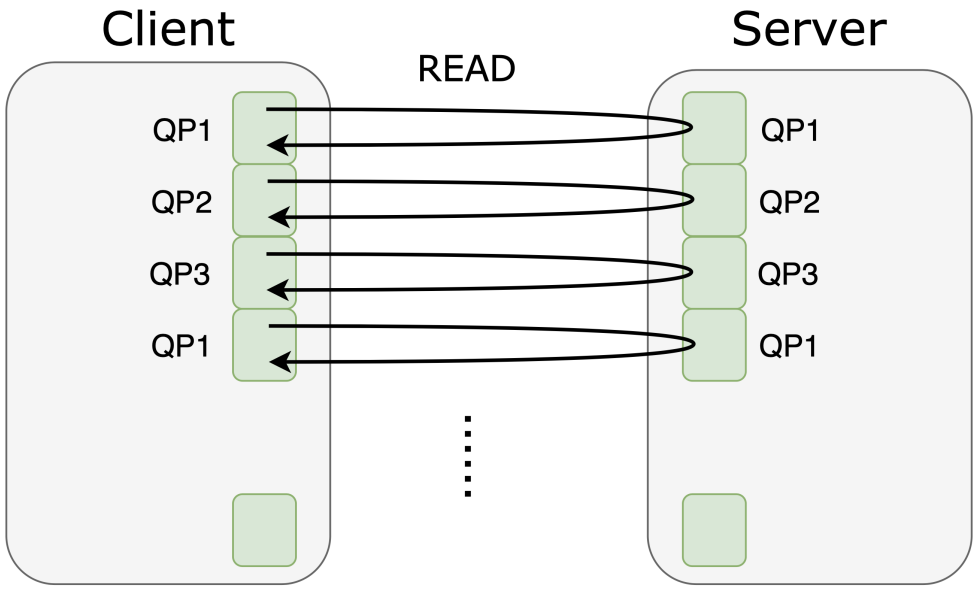
- Measure the actual minimum of the timeout
- Interesting enough, the timeout cannot be set to be any smaller than 500 ms
  - This minimal value is pre-configured in the firmware, and users have no means to modify it
- This configuration is not problematic when ODP is disabled

The second pitfall: packet flood

- Situation
  - Issue READs using multiple QPs with client-side ODP
- Characteristics
  - Huge number of packets by retransmission of requests
- Effect
  - Latency of several hundred milliseconds to several tens seconds
- Cause
  - Failure of updating page statuses among QPs

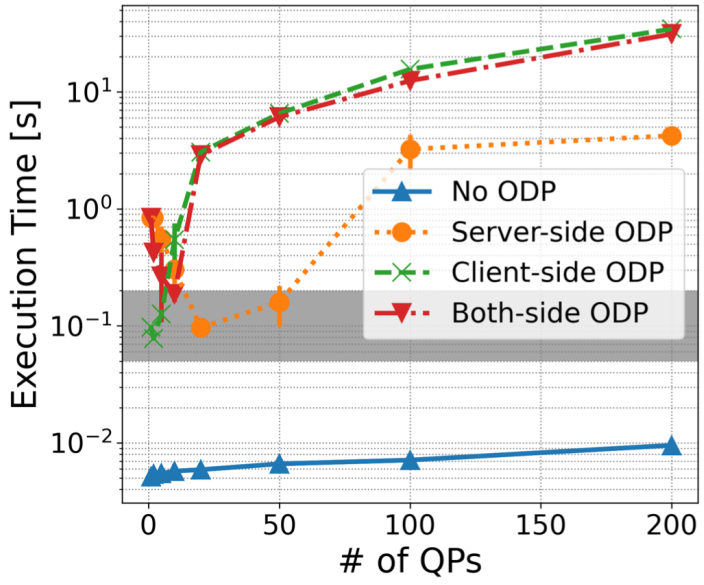


Microbench for packet flood

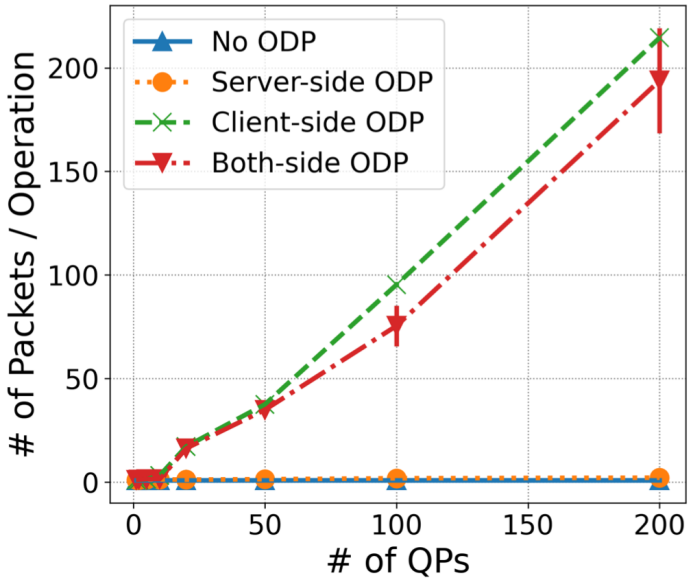


- 8192 READs with a message size of 100 bytes
- No intervals between communication operations
- Change the number of QPs

Execution time and # of packets



(a) Execution Time (s)



(b) Number of Packets

- Observe the super-long latency of several seconds and a huge number of packets with an increasing number of QPs
- The root cause: the failure of updating page statuses among QPs (see paper)

Summary

- Find two critical performance pitfalls of ODP
  - The latency is longer by 3–4 orders of magnitude than the overhead of a NIC's page fault itself
  - Reproduce them with simple microbenchmarks
- Take-home messages
  - Both pitfalls are related to concurrent page faults
  - Current H/W-based implementation of ODP is much more fragile than expected
- Future work: Better implementation of ODP
  - We have reported them to the vendor
  - Approach coordinated with S/W could be one choice