OPENFABRICS ALLIANCE

2013 OFA Developer Workshop

Network Direct v2 and WinOFED

Fab Tillier (ftillier@microsoft.com)

What is Network Direct?

- Microsoft defined interfaces for RDMA
- Transparent support of IB, iWARP, and RoCE
- IP-addressing based

Why Network Direct?

- Designed for Windows
- Higher level of abstraction
- Fabric agnostic
- Stable ABI
- No Callbacks
- Dynamic provider discovery
- Extensible
- Easier to understand
- Easier to develop for

Asynchronous Operations

- Win32 Overlapped operations used for:
  - Memory Registration
  - CQ Notification
  - Connection Management
- Client controls threading and completion mechanism
  - I/O Completion Port or GetOverlappedResult
- Simpler for kernel drivers to support
  - IoCompleteRequest – I/O manager handles the rest.

What's New in NDv2?

- Support for 3rd party applications
  - NDv1 was designed exclusively for MSMPI
- Alignment with NDK
- Easier extensibility
- Object hierarchy improvement
- Memory Regions are 1st class objects
- Shared Receive Queues
- CQ/SRQ Event Affinity
- Standards alignment

Using Overlapped I/O

IND2Adapter::CreateOverlappedFile

- Returns file handle on which overlapped requests are performed
- Client can bind it to its own I/O completion port
- Client can control I/O behavior
  - SetFileCompletionNotificationModes
    - FILE_SKIP_COMPLETION_PORT_ON_SUCCESS is always set
  - BindIoCompletionCallback
- Multiple overlapped files for NUMA binding

Limitations of Network Direct

- RDMA Atomic support
- Unreliable Datagram support
- Multicast support
- Send and Invalidate
- Immediate Data

NDv2 Availability

- Headers published with HPC Pack 2012 SDK
- Supported by MSMPI v4 (HPC Pack 2012)
- Upcoming support:
  - WinOFED 3.2 (May 2013)
  - Mellanox 4.40 (2Q2013)
  - [This Space For Rent]

Case Study: uDAPL

- IBAL offers technical and business limitations
  - No ABI versioning mechanisms
  - Limited IHV support
- uDAPL ported to Network Direct v2 interfaces
  - Stan Smith (Intel) did all the heavy lifting
  - Portability between 3rd party provider
- Test case for NDv2 outside of MSMPI
- Test case for viability of WinOFED as ULP distro

- Running dapltest over ND worked out of the box
  - But who runs dapltest for their business?

Case Study: Intel MPI

- Intel MPI uses uDAPL to interface to RDMA networks
  - Currently supported configuration is uDAPL over IBAL
- Test case for portability of uDAPL clients

- Intel MPI just worked, too!

- Results NOT endorsed by Intel
- DAPL/ND not yet performance tuned

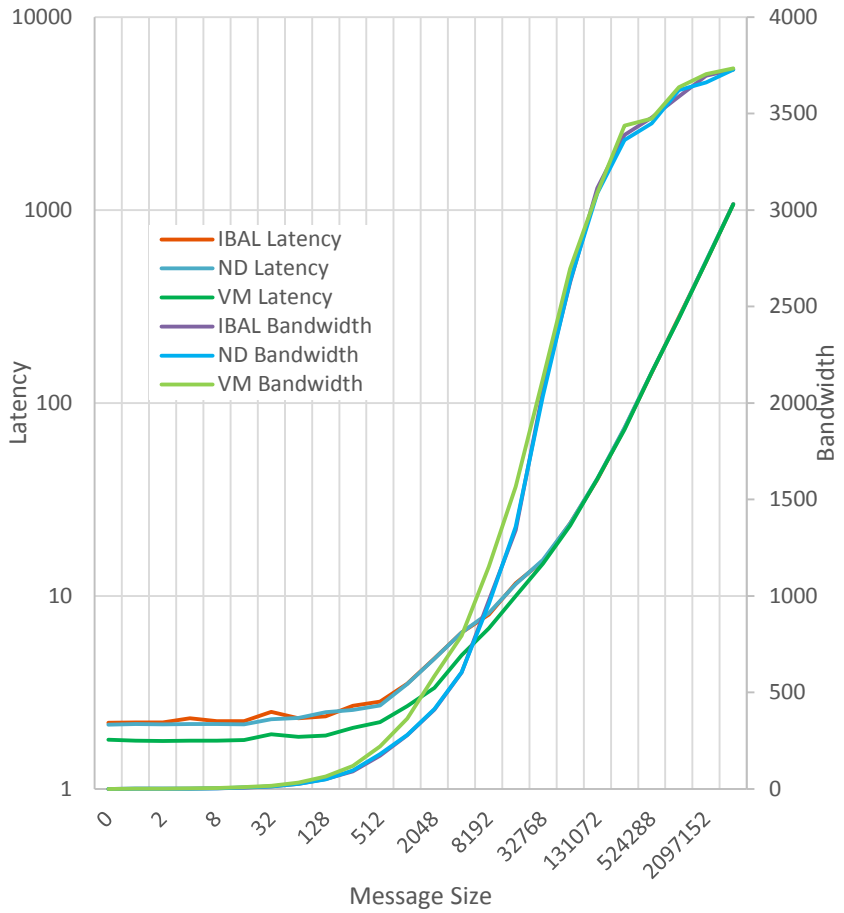
Intel MPI: Test Setup

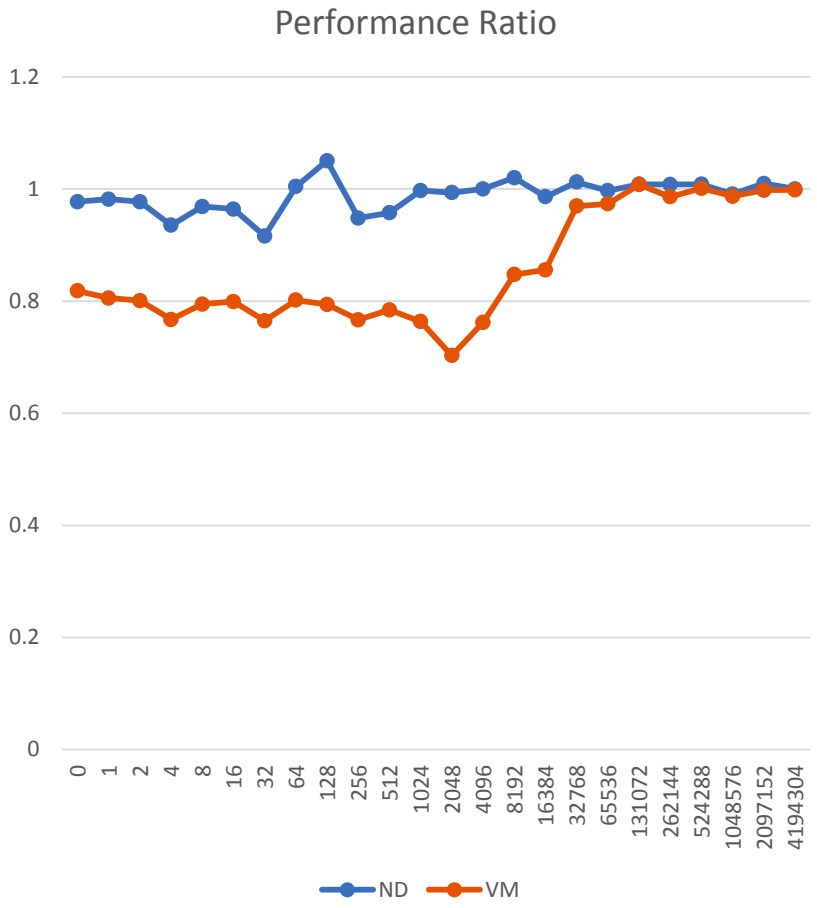
- Windows Azure Big Compute servers
  - Dual SandyBridge
  - Lots of RAM
  - QDR IB
  - Windows Azure OS
- Intel MPI Library, Version 4.0 Update 3 Build 20110824
- Intel MPI Benchmark Suite V3.2.3, MPI-1 part
  - Pingpong
  - Bcast
  - Allreduce
  - Reduce
  - Alltoall
  - Barrier

Intel MPI: Test Environment

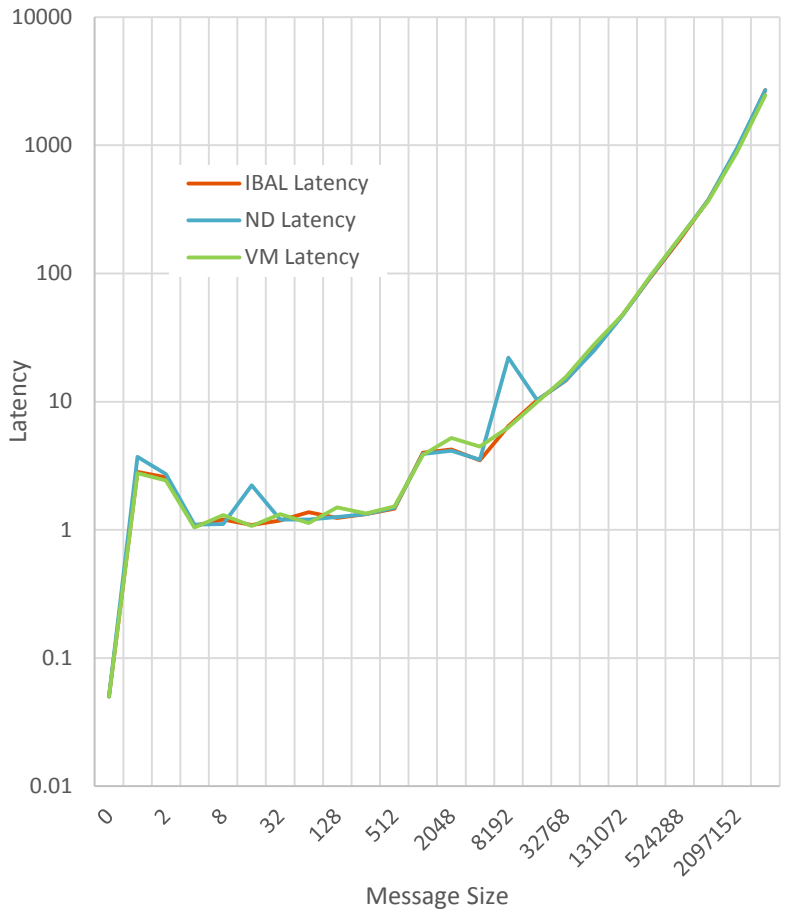
- Test environments
  - Two servers, one VM per server
  - Connected through IB switch
  - Windows Azure Host + IBAL
  - Windows Azure Host + NDv2
  - Windows Azure VM + NDv2
  - Single Run
    - no averaging or accounting for outliers

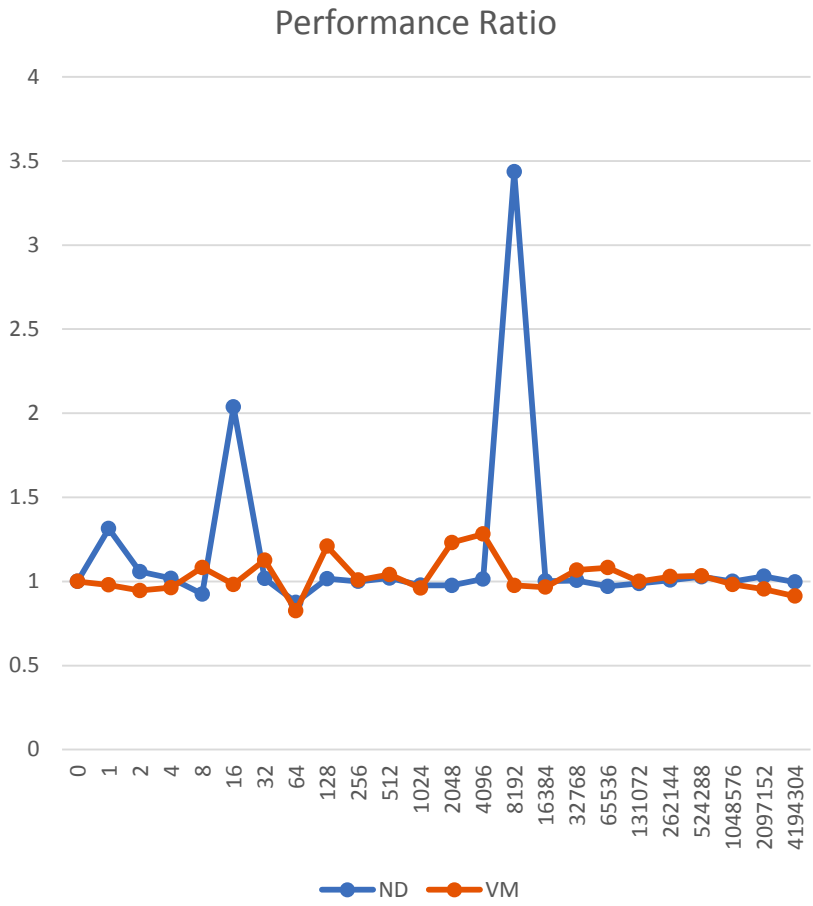
Pingpong


10000
1000
100
10
1
Latency
4000
3500
3000
2500
2000
1500
1000
500
0
Bandwidth
IBAL Latency
ND Latency
VM Latency
IBAL Bandwidth
ND Bandwidth
VM Bandwidth
0
2
8
32
128
512
2048
8192
32768
131072
524288
2097152
Message Size

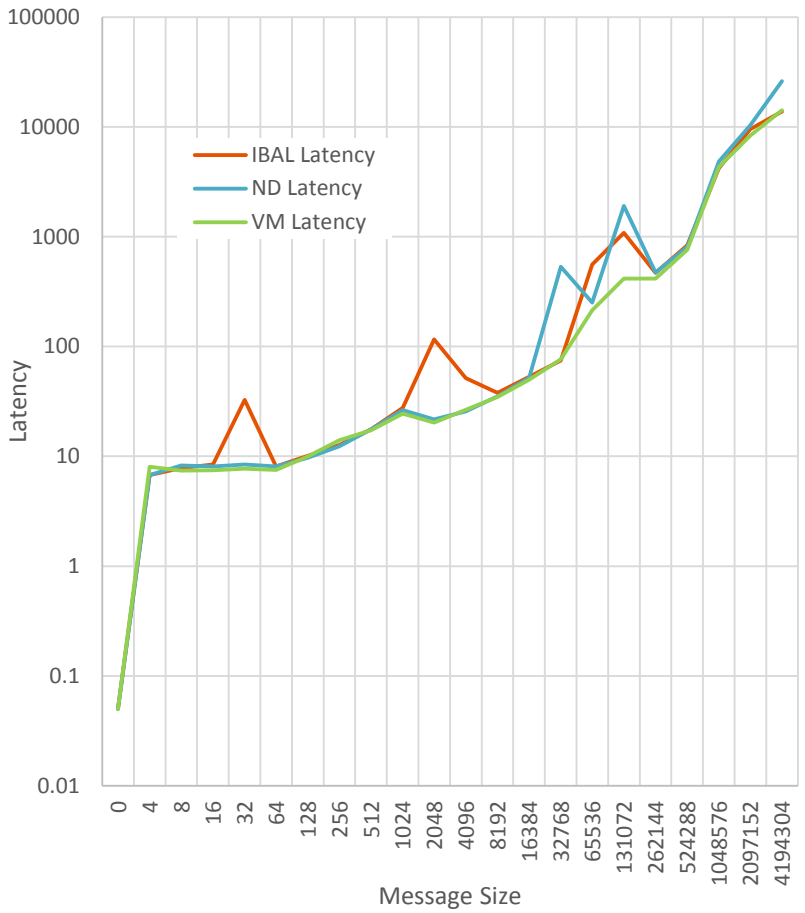

Performance Ratio
1.2
1
0.8
0.6
0.4
0.2
0
0
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
4194304
ND
VM

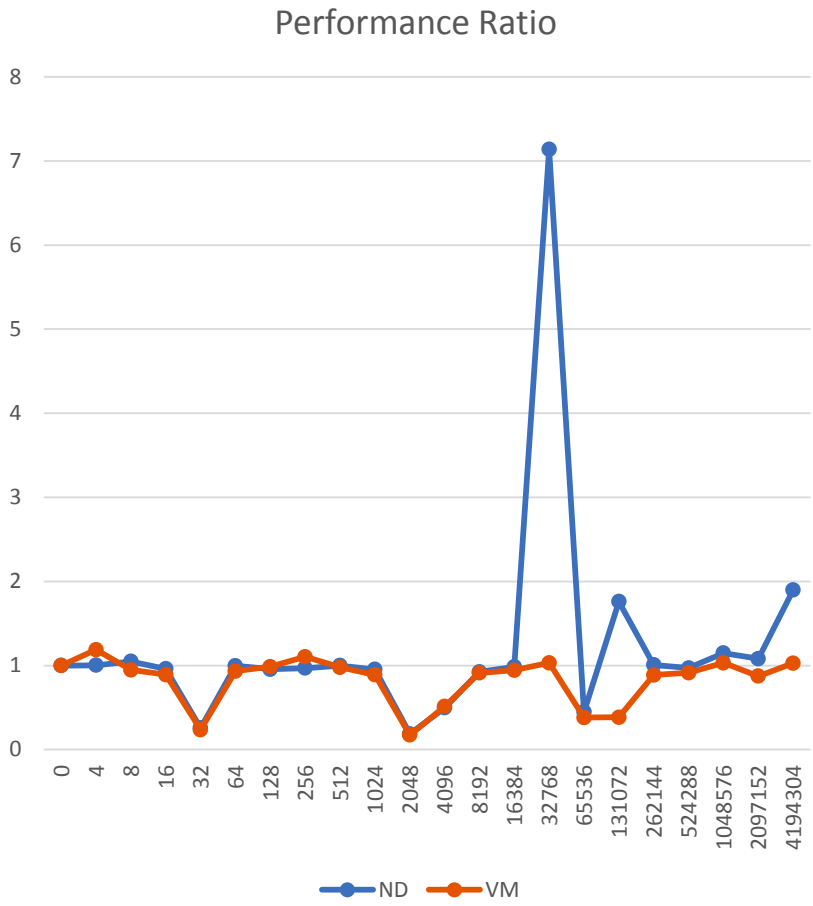
Bcast


10000
1000
100
10
1
0.1
0.01
Latency
IBAL Latency
ND Latency
VM Latency
0
2
8
32
128
512
2048
8192
32768
131072
524288
2097152
Message Size

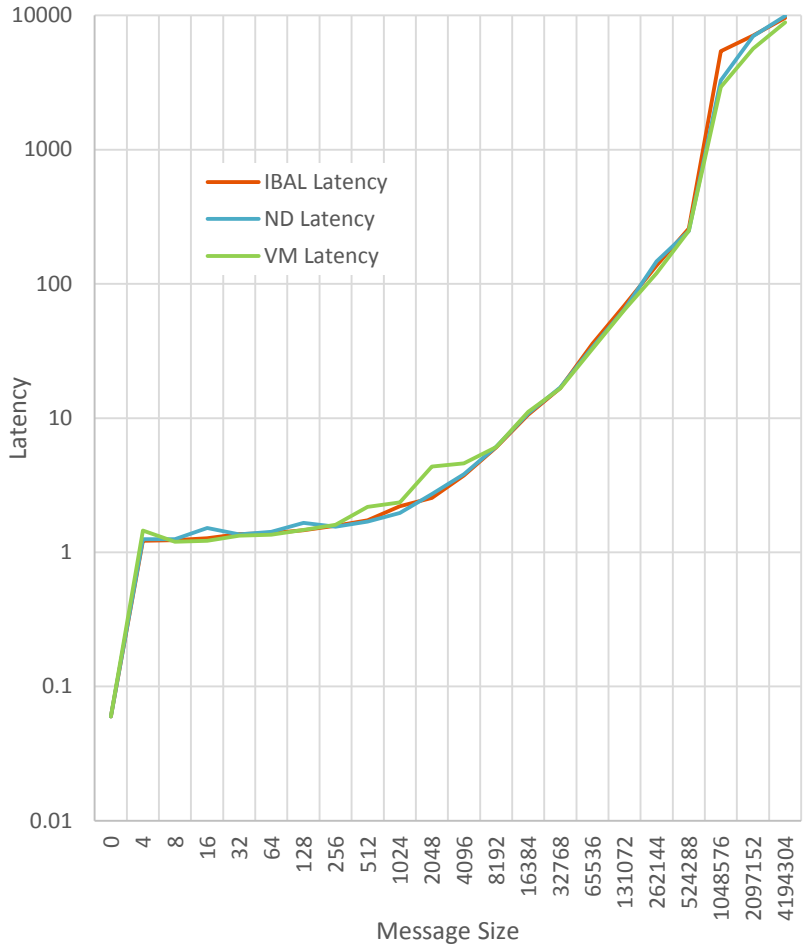

Performance Ratio
4
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
4194304
ND
VM

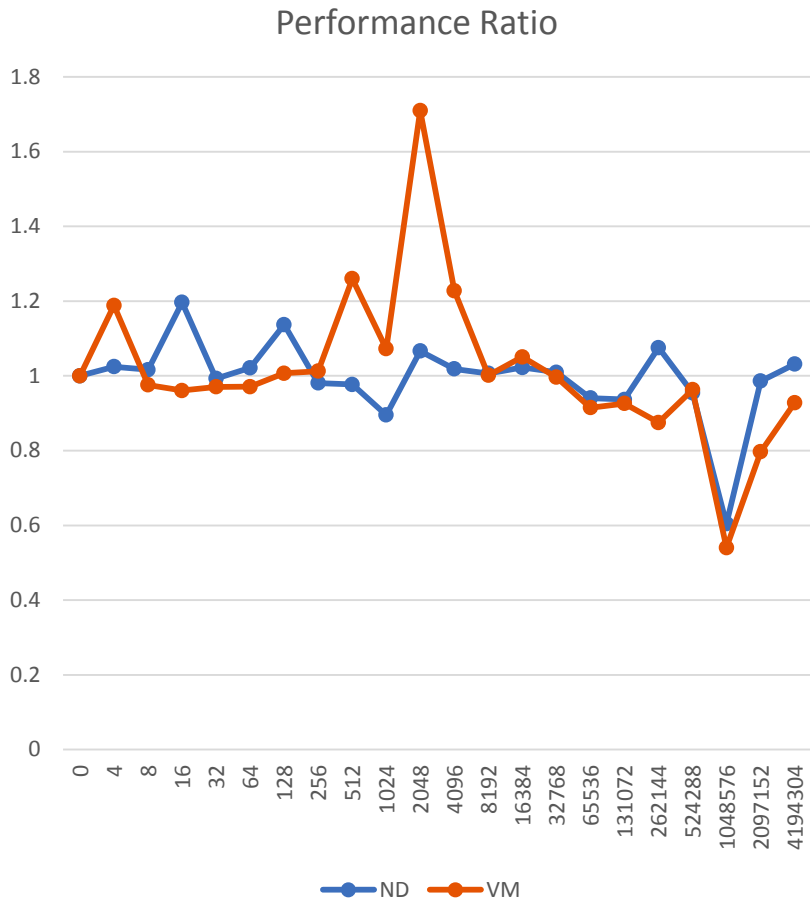
Allreduce


100000
10000
1000
100
10
1
0.1
0.01
Latency
IBAL Latency
ND Latency
VM Latency
0
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
4194304
Message Size

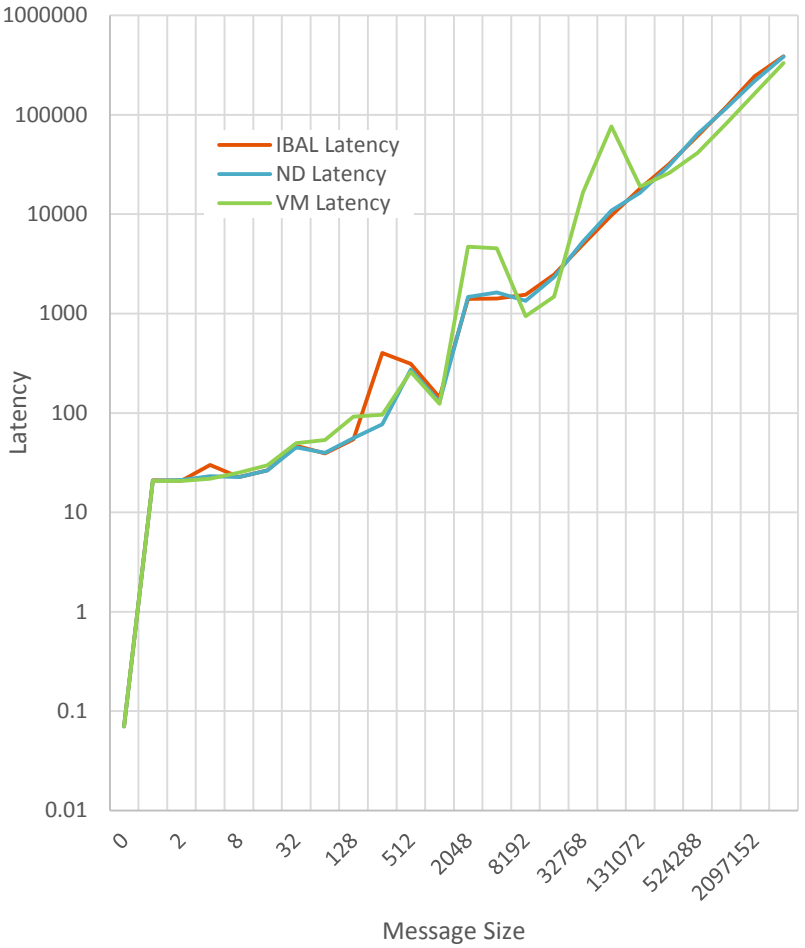

Performance Ratio
8
7
6
5
4
3
2
1
0
0
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
4194304
ND
VM

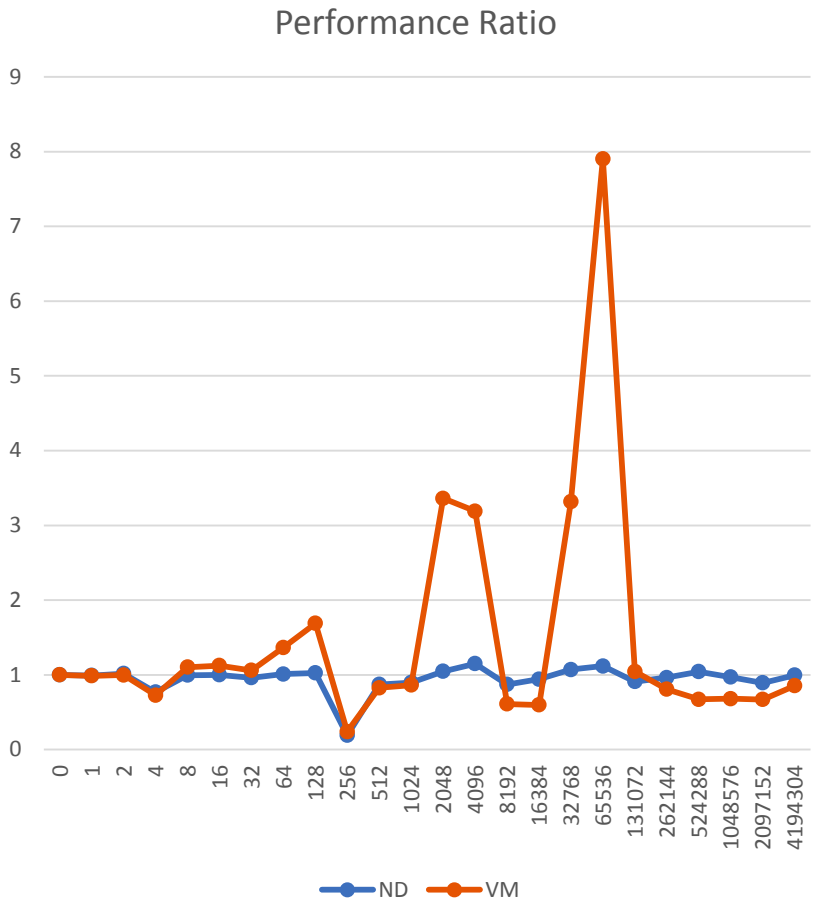
Reduce


10000
1000
100
10
1
0.1
0.01
Latency
IBAL Latency
ND Latency
VM Latency
0
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
4194304
Message Size

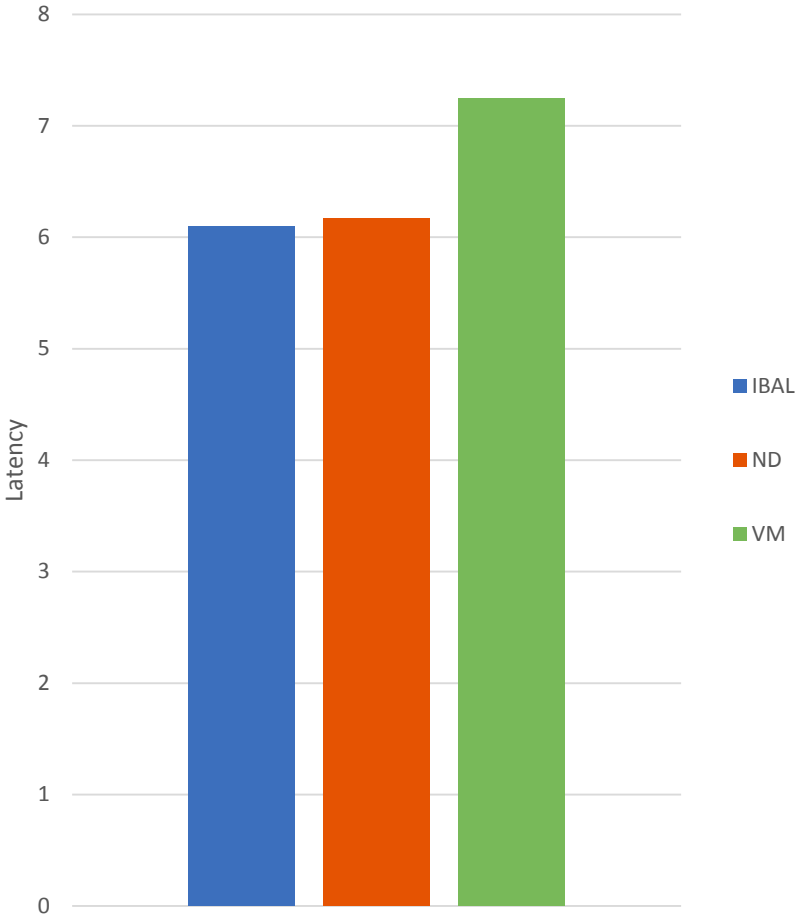

Performance Ratio
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
4194304
ND
VM

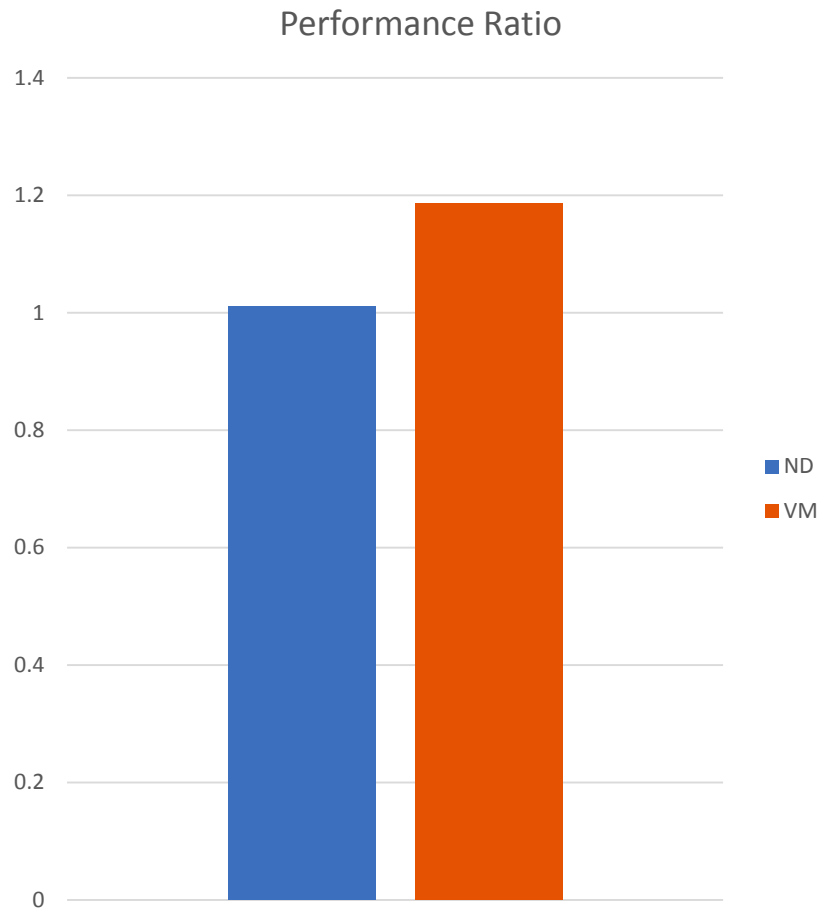
Alltoall


1000000
100000
10000
1000
100
10
1
0.1
0.01
Latency
IBAL Latency
ND Latency
VM Latency
0
2
8
32
128
512
2048
8192
32768
131072
524288
2097152
Message Size


Performance Ratio
9
8
7
6
5
4
3
2
1
0
0
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
4194304
ND
VM

Barrier


8
7
6
5
4
3
2
1
0
Latency
IBAL
ND
VM


Performance Ratio
1.4
1.2
1
0.8
0.6
0.4
0.2
0
ND
VM

Intel MPI: Summary

- ND performs competitively with IBAL
- No code changes required for running in Windows Azure Big Compute VMs

But…

- Needs further scale testing/tuning
- No support for UD ☹

WinOFED Value

- Instrumental in bringing RDMA support to Windows
- Key to prototyping efforts for Network Direct
- Benefits participating IHVs
- Proving ground for innovation
  - NDv1 provider published 2009
  - NDv2 provider published 2013

WinOFED Limitations

- Limited IHV support
  - Only Mellanox has participated in recent years
- Only IHVs can ship WHQL certified drivers
  - Effectively results in per-IHV (closed source) drivers
  - WinOFED participation not a WHQL requirement
- Spotty interoperability between WinOFED and IHV releases

WinOFED Future

- Declare mission accomplished?
- Find a way to engage more IHVs?
- Reduce the scope?
  - Build above well defined interfaces
    - Network Direct for Windows native code
    - Linux OFED for cross-OS portability
  - Multiple IHVs commit to support at the ABI level
  - Established forum for Microsoft participation

Call To Action

- Implement NDv2 in your Windows drivers
- Use NDv2 for your Windows RDMA apps
- Help build a richer ecosystem
  - Help redefine WinOFED
  - Broaden IHV participation
  - Take advantage of this inroad to Microsoft

Resources

- NDKPI Reference Documentation
- HPC Pack 2012 SDK

Thank You