

# **Azure DPDK Performance Report**

**March 2026**

DPDK 25.11.0

Microsoft Azure Network Adapter (MANA)

VM Size: Standard\_E192ids\_v6

Prepared by:  
Matthew G. McGovern  
Microsoft Linux Systems Group

# Index

**Cover - a**

**Index - b**

**Purpose - 1**

**Target Audience - 1**

**Azure Test Environment Description - 2**

- LISA Testing - 2
- Dedicated Adapters for DPDK - 2
- Note on Physical Proximity - 2
- VM Size and Capabilities - 3
- Performance Metrics – 3
- VM Isolation - 3

**Hardware Descriptions – 4..5**

- Accelerated Networking and SR-IOV – 4
- OpenHCL - 5

**CPU Layout and NUMA - 6**

**Test Description: testpmd – 7**

**Test Results: testpmd send/receive with multiple queues – 8**

**Further Reading – 9**

**Acknowledgements – 10**

## **Purpose**

The purpose of this document is to share basic performance data for DPDK on Microsoft Azure. The document covers Linux Virtual Machines (VMs) with the Microsoft Azure Network Adapter (MANA) on the Standard\_E192ids\_v6 offering. The document outlines the test environment, test cases and test data; as well as tools to reproduce the results.

## **Target Audience**

The intended audience of this document are software developers interested in evaluating the performance of DPDK on Azure. The reader should be familiar with the Data Plane Development Kit (DPDK), operating system (OS) virtualization and Linux.

## Azure Test Environment Description:

### Linux Integration Services Automation (LISA)

Microsoft's automated Azure testing for Linux leverages an open-source framework maintained by Microsoft: [Linux Integration Services Automation \(LISA\)](#). Knowledge of LISA is not required to reproduce the test results. A repository with shell scripts to prepare an environment and run the tests are available on [Github](#).

### Dedicated Network Adapter for DPDK

DPDK test cases on Azure require at least one dedicated network device; we recommend leaving one management NIC available while using DPDK on Azure. The tests in this report leave eth0 open and use eth1 for DPDK.

### Note on Physical Proximity

The physical location of a resource in Azure is normally determined by the 'region' or 'location' parameter. This location parameter corresponds to specific datacenters which vary in size. Customers can leverage the concept of 'availability sets' (a failover and data proximity abstraction in Azure) for physical proximity within a server rack. The test results in this report were recorded without leveraging any options to guarantee physical proximity. However, the VM size Standard\_E192ids\_v6 is guaranteed to ensure a single VM per server blade. The test VMs were created in the same Azure region within a few months of the release of the Standard\_E192ids\_v6 size. Some amount of physical proximity was a side effect of using this new size within a single region.

## **VM Size and Capabilities**

- VM SKU: Standard\_E192ids\_v6
- Cores: 192
- Memory: 1832GB
- Maximum NICs: 8
- Max Network Bandwidth: 200000 Mb/s
- [The full specifications for this size are available here.](#)

## **Performance Metrics**

The testpmd application measures sent, received and dropped packets. The application displays the data in packets-per-second (PPS). Packet size was set to the default value of 64 bytes.

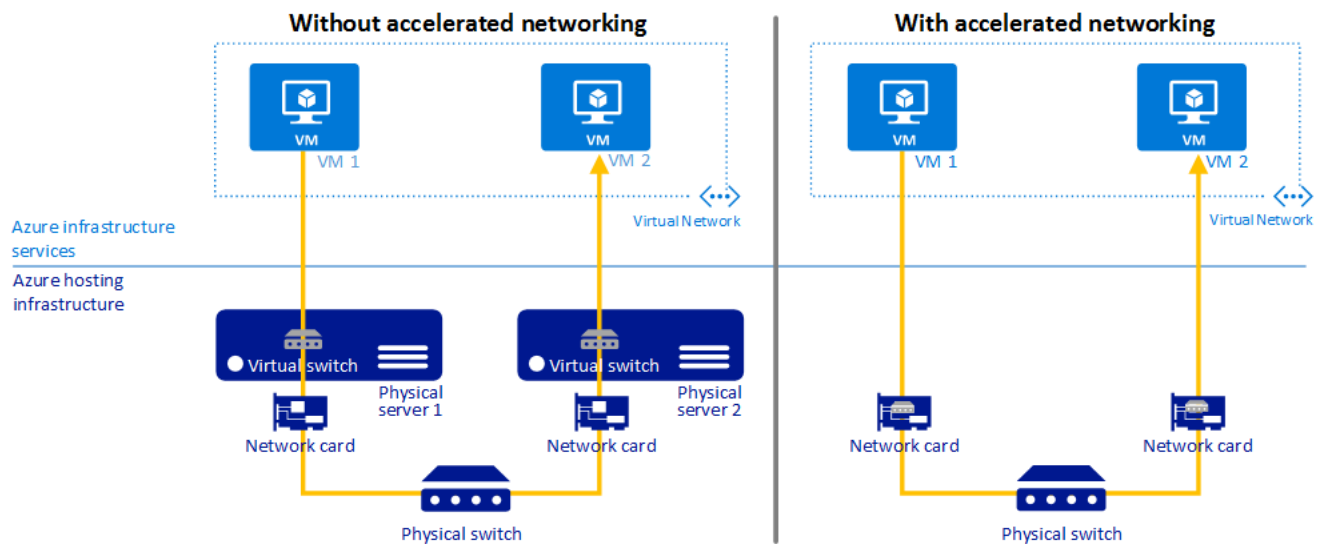
## **VM Isolation**

Standard\_E192ids\_v6 is an isolated VM size, it will never share a host with another VM. These PPS results are specific to this size. PPS tests on other VM families and VM sizes may result in lower PPS measurements. Maximum expected packet-per-second counts are hardware dependent and may change depending on the hardware and software that hosts a given VM.

## Hardware Descriptions:

### Azure Accelerated Networking and SR-IOV

Azure VMs can directly manage the hardware queues on a physical NIC via Single Root I/O Virtualization (SR-IOV). The result is higher throughput and lower latency than would be possible for an entirely virtualized network interface. Accelerated Networking was previously an optional feature for Azure VMs. Newer generations of Azure VM sizes require SR-IOV, so Accelerated Networking is always enabled for Standard\_E192ids\_v6 NICs.



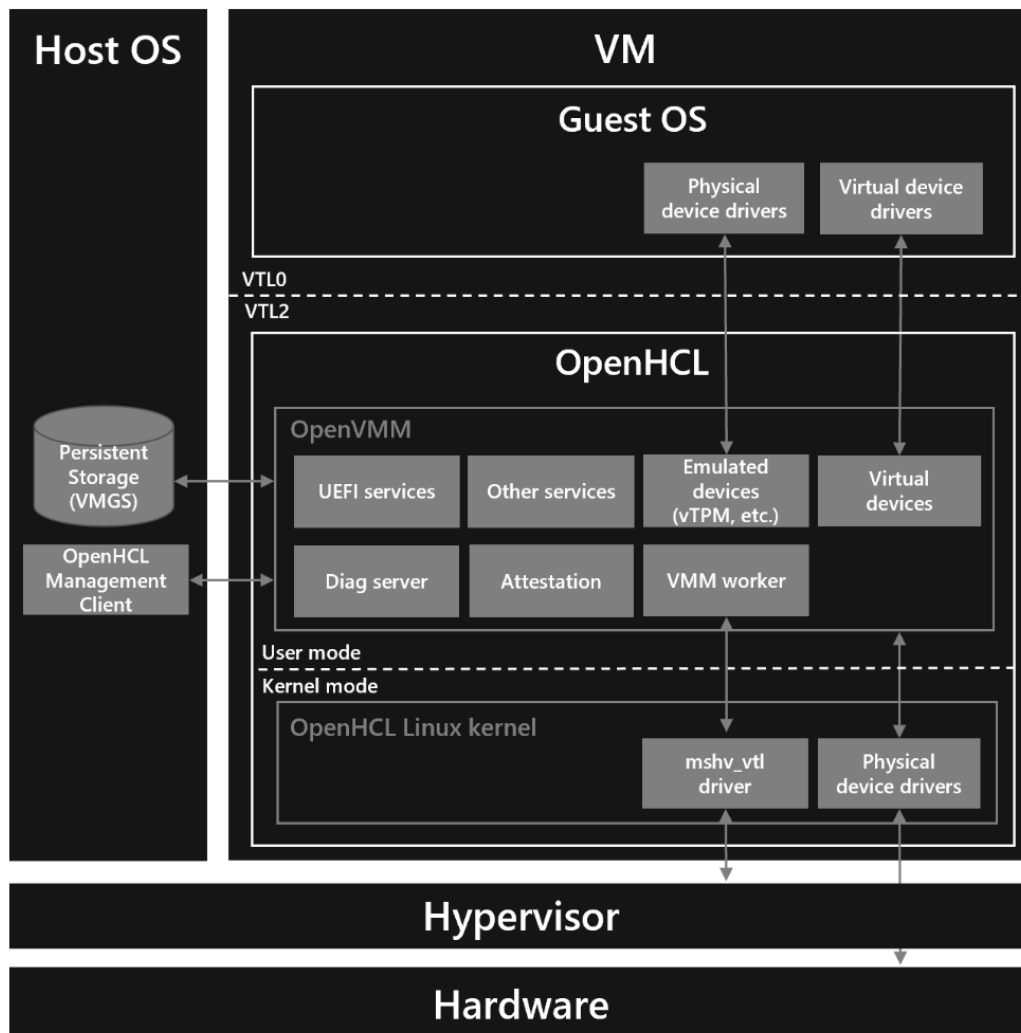
<https://learn.microsoft.com/en-us/azure/virtual-network/accelerated-networking-overview>

## OpenHCL

Azure Standard\_E192ids\_v6 VMs use an updated virtualization model compared to other offerings on Azure. Many services which required interaction with the Windows Host OS partition are now offloaded to an OpenHCL paravisor running within the guest partition. This change enables security and reliability features which are out-of-scope for this document.

Interested readers can find a high-level description on the [Microsoft Tech Blog: OpenHCL](#).

Both OpenHCL and OpenVMM are open source: [microsoft/openvmm on Github](#)



[OpenHCL: the new, open source paravisor | Microsoft Community Hub](#)

## CPU Layout and Non-Uniform Memory Access (NUMA):

This report provides data for the MANA NIC. The data is from a Standard\_E192ids\_v6 VM. [This size uses an Intel 5<sup>th</sup> Generation Xeon \(Emerald Rapids\) CPU.](#) Helpful information about the NUMA layout, CPU cores, caches and threads-per-core can be found with the `lscpu` application or the DPDK tool [dpdk/usertools/cpu\\_layout.py](#). Note that attempting to use multiple threads on the same physical core can degrade performance. Our tests aim to assign one DPDK lcore to each physical CPU by omitting certain core indices from the `testpmd` core list argument.

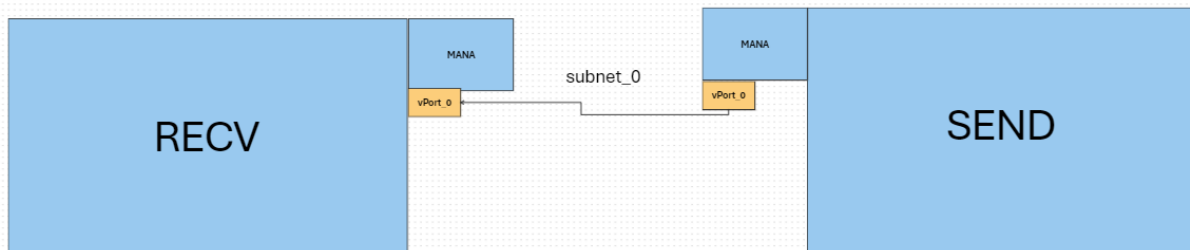
### lscpu

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:         48 bits physical, 57 bits virtual
Byte Order:            Little Endian
CPU(s):                192
On-line CPU(s) list:  0-191
Vendor ID:             GenuineIntel
Model name:            INTEL(R) XEON(R) PLATINUM 8573C
CPU family:            6
Model:                207
Thread(s) per core:   2
Core(s) per socket:   48
Socket(s):             2
Stepping:              2
CPU max MHz:          2300.0000
CPU min MHz:          800.0000
BogoMIPS:              4599.99
Virtualization features:
Virtualization:       VT-x
Hypervisor vendor:    Microsoft
Virtualization type:  full
Caches (sum of all):
L1d:                  4.5 MiB (96 instances)
L1i:                  3 MiB (96 instances)
L2:                   192 MiB (96 instances)
L3:                   520 MiB (2 instances)
NUMA:
NUMA node(s):         4
NUMA node0 CPU(s):   0-47
NUMA node1 CPU(s):   48-95
NUMA node2 CPU(s):   96-143
NUMA node3 CPU(s):   144-191
```



## Test Description:

### dpdk-testpmd:



A Sender VM sends traffic to a Receiver VM using dpdk-testpmd. The Sender VM sets the testpmd `--tx-ip` flag to the destination address. Packet size is left as the default (64 bytes, UDP).

### testpmd invocation notes:

Our tests use the testpmd `--tx-ip` flag to define the IP address of the source and destination vNICs. This flag is used because Azure Virtual Networks (vNets) will not route based on the MAC address alone. The vNet configuration and any associated subnet and virtual routing table rules must be configured for traffic to flow. Packets generated by testpmd must have an IP address and port number that satisfies the vNet configuration rules to successfully traverse the vNet, and those rules are defined for layer 3.

The `--txonly-multi-flow` flag is enabled to spread traffic across multiple UDP ports; this ensures that there is enough entropy in the packet UDP headers for receive-side-scaling (RSS) to hash the packets and spread them across multiple queues. Data for queue counts 1, 2, 4 and 8 are shown for comparison. The forwarding core count was set based on the number of queues.

## testpmd send/receive with multiple queues

**Forwarding vCPUs:** 8

**Hugepages on NUMA 0 (2MB):** 1024

**Network Adapter:** Microsoft Azure Network Adapter

**CPU Type:** Intel 5<sup>th</sup> Generation Xeon (Emerald Rapids)

**Azure VM SKU:** Standard\_E192ids\_v6

BIOS Hyper-V UEFI Release v4.1 07/25/2025

Hyper-V: Host Build 10.0.26102.1212-1-0

**Linux Kernel Version:** Ubuntu 25.04 (GNU/Linux 6.14.0-1017-azure x86\_64)

**Azure Marketplace Image:** Canonical:ubuntu-25\_10:server:25.10.202601080

**RDMA Core Version:** 59.0

**DPDK Version:** 25.11.0

**DPDK Queue Count:** 4 sender, 1-8 receiver

**Tools to reproduce:** [dpdk-perf: DPDK performance report data and scripts](#)

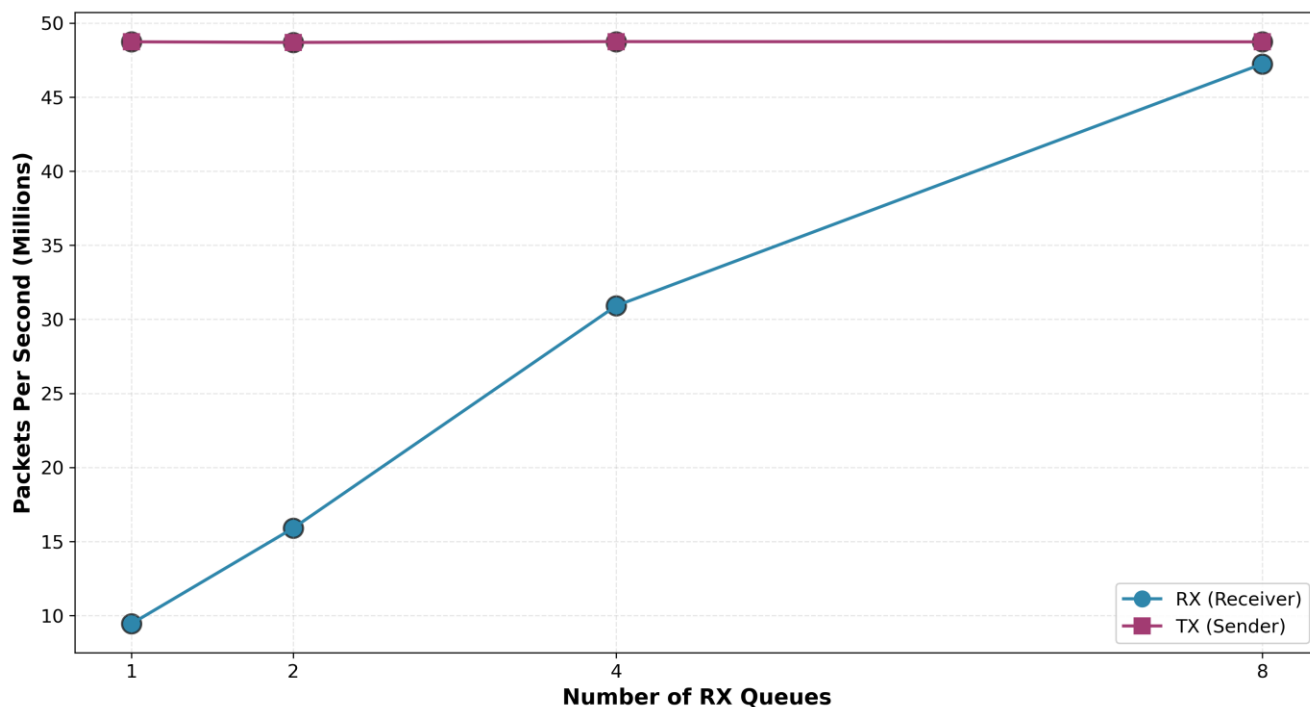
**Description:** Test DPDK PPS for multiple receiver queue counts.

### Application invocations:

```
sudo dpdk-testpmd -l 0,1,2,3,4 --vdev=7870:00:00.0,mac=$mac -- --forward-mode=txonly --txq=4 --rxq=4 --txd=256 --rxd=256 --stats-period=5 --nb-cores=4 --tx-ip=192.168.0.4,192.168.0.5 --txonly-multi-flow
```

```
sudo dpdk-testpmd -l $core_list --vdev=7870:00:00.0,mac=$mac -- --forward-mode=rxonly --txq=$queues --rxq=$queues --txd=1024 --rxd=1024 --stats-period=5 --nb-cores=$core_count
```

**Network Performance: RX vs TX PPS across RX Queue Configurations**



## Further reading

- [Data Plane Development Kit @ dpdk.org](https://dppk.org)
- [DPDK MANA Poll Mode Driver Documentation](#)
- [Azure Documentation: DPDK on Linux with MANA](#)
- [Linux Kernel: mana\\_ib driver source](#)
- [rdma-core: mana provider](#)
- [Azure Accelerated Networking documentation](#)

## **Acknowledgements:**

Thanks to the DPDK Project, Linux Foundation, Microsoft Linux Systems Group, Azure Linux, Azure Boost, Azure Host Networking and Network Performance groups; as well as the OpenHCL contributors.

Additional personal thanks (in no specific order) to Khrum Kashan Jat, Long Li, Wei Hu, Haiyang Zhang, Dexuan Cui, Boqun Feng, Matt Reat, Brian Denton, Paul Rosswurm, Sharath George John, Margaret Sands, Avi Levy, Shradha Gupta, Konstantin Taranov and John Bruner for their help along the way.

Questions? Concerns? Found a bug?

[dpdk@microsoft.com](mailto:dpdk@microsoft.com)