CLAIMS

1. A method, by a processor, for utilizing an enhanced network stack framework in a computing environment, comprising:

sharing, in the enhanced network stack framework, a plurality of network buffers coherently attached between one or more applications and a physical network interface of a non-virtualized computing system while bypassing one or more drivers and an operating system of the non-virtualized computing system, wherein the sharing of the plurality of network buffers is executed by a network stack sharing service inclusive of an application buffer, a circular buffer and a queuing and pooling operation, wherein the plurality of network buffers coherently attached between the one or more applications provide a shared library providing a shared region and a private region to the one or more applications, and wherein the one or more applications exchange in-memory data over the physical network interface using a simplified remote direct memory access (RDMA)-style network communications exclusively implementing pointers and spinlocks to reduce RDMA round trip latency.

2. The method of claim 1, further including controlling the plurality of network buffers by a shared library.

3. The method of claim 1, further including sharing one or more address spaces of the plurality of network buffers between the one or more applications using the physical network interface, wherein the plurality of network buffers used for input/output (I/O) control.

4. The method of claim 1, further including exchanging memory pointers with coherently attached devices using the circular buffer.

5. The method of claim 1, further including executing the queuing and pooling operation for the plurality of network buffers for network buffer transmission, reception, and manipulation.

6. The method of claim 1, wherein the queuing and pooling operation further includes moving, assigning, or reassigning one of the plurality of network buffers from one or more queues and one or more pools.

7. A system for utilizing an enhanced network stack framework, comprising:

one or more computers with executable instructions that when executed cause the system to:

share, in the enhanced network stack framework, a plurality of network buffers coherently attached between one or more applications and a physical network interface of a non-virtualized computing system while bypassing one or more drivers and an operating system of the non-virtualized computing system, wherein the sharing of the plurality of network buffers is executed by a network stack sharing service inclusive of an application buffer, a circular buffer and a queuing and pooling operation, wherein the plurality of network buffers coherently attached between the one or more applications provide a shared library providing a shared region and a private region to the one or more applications, and wherein the one or more applications exchange in-memory data over

the physical network interface using a simplified remote direct memory access (RDMA)-style network communications exclusively implementing pointers and spinlocks to reduce RDMA round trip latency.

8. The system of claim 7, wherein the executable instructions when executed cause the system to control the plurality of network buffers by a shared library.

9. The system of claim 7, wherein the executable instructions when executed cause the system to share one or more address spaces of the plurality of network buffers between the one or more applications using the physical network interface, wherein the plurality of network buffers used for input/output (I/O) control.

10. The system of claim 7, wherein the executable instructions when executed cause the system to exchange memory pointers with coherently attached devices using the circular buffer.

11. The system of claim 7, wherein the executable instructions when executed cause the system to execute the queuing and pooling operation for the plurality of network buffers for network buffer transmission, reception, and manipulation.

12. The system of claim 7, wherein the executable instructions when executed cause the system to move, assign, or reassign one of the plurality of network buffers from one or more queues and one or more pools for executing the queuing and pooling operation.

13. A computer program product for utilizing an enhanced network stack framework by a processor, the computer program product comprising a non-transitory computer-readable storage medium having program instructions embodied thereon, the program instructions executable by a processor to cause the processor to:

share, in the enhanced network stack framework, a plurality of network buffers coherently attached between one or more applications and a physical network interface of a non-virtualized computing system while bypassing one or more drivers and an operating system of the non-virtualized computing system, wherein the sharing of the plurality of network buffers is executed by a network stack sharing service inclusive of an application buffer, a circular buffer and a queuing and pooling operation, wherein the plurality of network buffers coherently attached between the one or more applications provide a shared library providing a shared region and a private region to the one or more applications, and wherein the one or more applications exchange in-memory data over the physical network interface using a simplified remote direct memory access (RDMA)-style network communications exclusively implementing pointers and spinlocks to reduce RDMA round trip latency.

14. The computer program product of claim 13, wherein the program instructions executable by the processor further cause the processor to control the plurality of network buffers by a shared library.

15. The computer program product of claim 13, wherein the program instructions executable by the

processor further cause the processor to share one or more address spaces of the plurality of network buffers between the one or more applications using the physical network interface, wherein the plurality of network buffers used for input/output (I/O) control.

16. The computer program product of claim 13, wherein the program instructions executable by the processor further cause the processor to: exchange memory pointers with coherently attached devices using the circular buffer; and
execute the queuing and pooling operation for the plurality of network buffers for network buffer transmission, reception, and manipulation.

17. The computer program product of claim 13, wherein the program instructions executable by the processor further cause the processor to move, assign, or reassign one of the plurality of network buffers from one or more queues and one or more pools for executing the queuing and pooling operation.