

CLAIMS

1. A method, by a processor, for utilizing an enhanced network stack framework in a computing environment, comprising:
sharing a plurality of network buffers coherently attached between one or more applications and a network interface while bypassing one or more drivers and an operating system using an application buffer, a circular buffer and a queuing and pooling operation.
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 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. The method of claim 1, further including establishing a shared memory region and a private memory region using the plurality of network buffers.
8. 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 a plurality of network buffers coherently attached between one or more applications and a network interface while bypassing one or more drivers and an operating system using an application buffer, a circular buffer and a queuing and pooling operation.
9. The system of claim 8, wherein the executable instructions when executed cause the system to control the plurality of network buffers by a shared library.

10. The system of claim 8, 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 network interface, wherein the plurality of network buffers used for input/output (I/O) control.

11. The system of claim 8, wherein the executable instructions when executed cause the system to exchange memory pointers with coherently attached devices using the circular buffer.

12. The system of claim 8, 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.

13. The system of claim 8, 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.

14. The system of claim 8, wherein the executable instructions when executed cause the system to establish a shared memory region and a private memory region using the plurality of network buffers.

15. 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 computer-readable program code portions stored therein, the computer-readable program code portions comprising: an executable portion that shares a plurality of network buffers coherently attached between one or more applications and a network interface while bypassing one or more drivers and an operating system using an application buffer, a circular buffer and a queuing and pooling operation.

16. The computer program product of claim 15, further including an executable portion that controls the plurality of network buffers by a shared library.

17. The computer program product of claim 15, further including an executable portion that shares one or more address spaces of the plurality of network buffers between the one or more applications using the network interface, wherein the plurality of network buffers used for input/output (I/O) control.

18. The computer program product of claim 15, further including an executable portion that: exchanges memory pointers with coherently attached devices using the circular buffer; and executes the queuing and pooling operation for the plurality of network buffers for network buffer transmission, reception, and manipulation.

19. The computer program product of claim 15, further including an executable portion that moves, assigns, or reassigns one of the plurality of network buffers from one or more queues and one or more pools for executing the queuing and pooling operation.

20. The computer program product of claim 15, further including an executable portion that establishes a shared memory region and a private memory region using the plurality of network buffers.