1 Evaluation

In this section, we evaluate the uniRDMA prototype on the physical RDMA platform. In addition to the basic network performance, such as throughput and latency, it also includes real-wprd RDMA applications. We expect to answer the following questions: (1)Can uniRDMA's network performance in container and virtual machine environments be close to native RDMA? (2)Does uniRDMA have high scalability in both large-scale container and virtual machine cluster environments? (3)Can uniRDMA be adapted to the real RDMA application environment in containers and virtual machines?

1.1 Experiment Set Up

All experiments are carried out on two servers. The settings mainly include three parts: host server, container and virtual machine. The detailed settings are shown in Table 6-1:

The RNIC used by the server is Mellanox ConnectX-3 56 Gb/sec, which performs RDMA communication under Infiniband. The operating systems of servers, containers, and virtual machines are all CentOS 7.4.1708, and the corresponding Linux kernel version is 3.10.0-693.el7.x86_64. The RDMA driver installed on the host server is Mellanox OFED 4.4-2.0.7.0, which adapts to the RNIC and host operating system. In addition, the entire uniRDMA framework is compiled with GCC/G++ 4.8.5, and the O3 compilation optimization level is selected.

1.2 Basic benchmark

Throughput and latency are the key target of network performance. RDMA supports two different data transmission modes: unilateral and bilateral. Due to the difference performance between thems, we evaluate them respectively.

Based on the RDMA benchmark test tool Mellanox perftest, we evaluated the throughput and latency of uniRDMA, native RDMA, hardware virtualization SR-IOV, and software virtualization FreeFlow in virtual machines or containers. For bilateral operations (Send and Recv), we use the "ib_send_bw" and "ib_send_lat" commands; for two unilateral operations (Write and Read), with Write as the representative, we use the "ib_write_bw" and "ib_write_lat" commands. The specific process is: after the RDMA connection is established between the client and the server, the bytes of transmitted message each time will be increased from 4B to 1MB, the data will be iteratively transmited 1000 times with each message size, and finally the average throughput and latency are calucated.

(1)Throughput: The results of bilateral operation are shown in Figure 6-1, and the one of unilateral operation are shown in Figure 6-2. Whether uniRDMA is in a virtual machine or in a container scenario, the throughput of its bilateral and unilateral operations is slimilar as SR-IOV and close to native RDMA.

Compared with FreeFlow, when the message is small, the throughput of uniRDMA has reached 4-6 times that of FreeFlow. Because FreeFlow forwards all data commands to the software virtualization layer for processing. Therefore, the forward latency gradually accumulates and decrease the throughput significantly. However, uniRDMA maps all RDMA resources to execute data commands in the user space of the container or virtual machine. Therefore, there is no latency for commands forwarding in data path.

When the message gradually increases, such as reaching 64KB, the throughput of each framework tends to be consistent. The reason is that the bandwidth is saturated, and the delay overhead of FreeFlow has been covered by waiting delay in RNIC.

(2)latency: The results of bilateral operation are shown in Figure 6-1, and the one of unilateral operation are shown in Figure 6-2. Whether uniRDMA is in a virtual machine or in a container scenario, the latency of its bilateral and unilateral operations is slimilar as SR-IOV and close to native RDMA.

Compared with FreeFlow, when the message is small, the latency of uniRDMA has reached 40% 60% of FreeFlow beacause of FreeFlow's forwarding latency. Also, when the message gradually increases, such as reaching 64KB, the latency of each framework tends to be consistent. Beacause the main latency has been caused by RNIC data processing.

(3)Scabality: Scalability is a challenge that RDMA virtualization needs to face under large-scale virtual machine or container clusters. In order to analyze the scalability of uniRDMA, this article uses the "ib_write_bw" command in Mellanox perftest to test uniRDMA between two servers. For 2, 4, 8, 16, 32, 64, and 128 pairs of virtual instances, a random number of virtual instances are selected. Virtual instance pair, execute this command to test the throughput result of sending 128KB data. Among them, for virtual instances of servers, there are scenarios where all virtual machines and all containers are used, and there are also scenarios where virtual machines and containers are mixed (50% each). Finally, the average throughput between virtual instance pairs is shown in Figure 6-5.

From Figure 6-5, in the full virtual machine scenario, full container scenario, and virtual machine and container mixed scenario, uniRDMA has good scalability and still maintains a stable RDMA throughput performance. In contrast, the throughput of FreeFlow at this time is less than 10% of the throughput of uniRDMA. Because uniRDMA data commands do not need to be forwarded to the software virtualization layer for processing, the virtualization overhead does not increase due to the expansion of virtual instances. When FreeFlow runs RDMA commands at the same time in a large-scale container, it is forwarded to the same RDMA context of the routing virtual layer for processing. When calling the QP queue or data block corresponding to the virtual instance data command, there is the overhead of lock mutual exclusion. Therefore, FreeFlow suffers from a drastic drop in performance in large-scale container scenarios.

In addition, as shown in Figure 6-5, both uniRDMA and FreeFlow can support communication between 128 pairs of virtual instances. However, the maximum number of VFs of the Mellanox ConnectX-3 network card is only 126, so 128 pairs of virtual instances are not supported. This shows that uniRDMA has higher scalability than SR-IOV. Because: SR-IOV technology statically allocates the VF interface to a virtual machine, which is exclusively occupied by the virtual machine, and cannot be dynamically shared; while uniRDMA improves the utilization of VF through the virtual layer dynamic device pool and flexible mapping mechanism. Can meet the virtual instance scale exceeding the number of VFs.

1.3 Real-world Applications

The high performance of the RDMA network needs to provide effective performance improvements for various big data applications in order to reflect its value. A good RDMA virtualization framework needs to have performance effects close to native RDMA applications in various scenarios. Therefore, in order to test the performance of uniRDMA in real RDMA application scenarios, this paper tested the performance results of uniRDMA and other frameworks in high-performance computing applications Graph-500, big data framework Spark and other RDMA applications.

(1)Graph-500: In the field of high-performance computing, Graph-500 is a benchmark framework used to test the performance of the Message Passing Interface (MPI) [42]. Based on the constructed graph structure, users test the performance of breadth-first search (BFS) and single source shortest path (SSSP). The performance index is the number of edges traversed per second (traversed edges). per second, TEPS), the larger the value, the better the performance.

In this paper, the node scale of the computational graph in Graph-500 is set to 26, and the ratio of edges to points is set to the default parameter of 16. The constructed graph has a total of $2\hat{2}5$ vertices, with $2\hat{2}9$ edges, the entire graph occupies approximately Around 16GB. When testing BFS and SSP, 16 MPI processes are scattered and executed on two nodes in turn, and the average value is taken according to the results of 12 tests. The data obtained is shown in Table Figure 6-6 (because FreeFlow is testing During the process, there was a program crash problem, so the corresponding data is lacking).

It can be seen from Figure 6-6 that the performance of uniRDMA is equivalent to that of host RDMA and hardware virtualization SR-IOV technology, and there is no obvious performance loss. This is because uniRDMA bypasses the kernel and virtualization layer in the data path, and there is no software forwarding delay caused by virtualization.