1 Background

This section introduces the background knowledge related to RDMA virtualization, including the principles and working mechanisms of traditional network virtualization and RDMA networks.

1.1 Traditional Network Virtualization

In the traditional network virtualization process, virtual network cards and virtual bridges are needed to realize network connection and data transmission between virtual instances. When two virtual instances communicate across nodes, the traffic of the virtual network card will be forwarded to the physical network card through the virtual bridge, and then through the remote physical network card and virtual bridge to enter the destination virtual instance.

The main implementation method in the virtual network card is software simulation. According to the implementation mode of the virtual network card and the applicable scenario, it can be divided into full virtualization, paravirtualization and container virtual network card. They can all be configured with corresponding virtual IP addresses and can simulate the basic functions of the network card to send and receive data packets. Traditional network virtualization also requires the use of virtual network bridges, which connect virtual network cards and physical network cards through routing and forwarding, tunnel networking, and other methods. Through the unified management of virtual network cards and virtual bridges, a complete virtual network is constructed.

1.2 RDMA network

RDMA is currently widely used in fields such as artificial intelligence, big data processing, and high-performance computing. Frameworks such as TensorFlow, Spark, and Hadoop [21] all have corresponding RDMA versions. Through hardware protocol stack and zero copy technology, RDMA network card can bypass the kernel to read and write remote memory data according to the work request of the application, without the participation of remote CPU. Therefore, RDMA has high throughput, low latency and low The network performance of the CPU load.

In order to take advantage of the high performance of the RDMA network, applications need to access and control the RDMA network card with the help of the Verbs interface. The RDMA network card and the Verbs interface provide applications with a communication method based on Queue Pair (QP). The application writes the RDMA work request to the QP, and then performs a write operation on the doorbell register of the network card, and informs the network card processor to execute the work request in the QP to transfer data. The entire process can be implemented by calling the Verbs interface in the user space without going through the kernel.

The QP queue consists of a pair of Send Queue (SQ) and Receive Queue (RQ), which respectively serve the sending and receiving work requests during RDMA data transmission. After the RDMA connection is established, the RDMA network card and the Verbs interface support data transmission for bilateral operation and unilateral operation:

- `Bilateral operation`: Both ends of the RDMA connection execute the sending work request and the receiving work request respectively, that is, each writes the sending or receiving work request into the corresponding SQ or RQ queue. Similar to the traditional TCP network, the sender's RDMA network card will transmit data to the receiver. At this time, the participation of the CPUs at both ends is required.
- `Unilateral operation`: Only one end of the RDMA connection needs to perform the sending work request. The RDMA application writes and sends work requests to the SQ queue in the QP, and the RDMA network card will read and write the remote memory according to the request without the participation of the remote CPU. 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.

```
\documentclass[sigplan,screen]{acmart}
```