UniRDMA Design and Implementation

1. Introduction

RDMA are high-performance network

Cloud are popular and hybrid

RDMA virtualization unified goals with performance and manageability

Hardware virtualization cannot achieve

Software virtualization cannot achieve

So, provide uniRDMA

Address challenges:

* VM and container are different virtualization v.s. Unified, centralized RDMA virtual layer
* VM‘s and container’s RDMA APP are different v.s. Unified RDMA interface

To address above:

* A unified centralized virtual layer and maintain container’s lightweight
* A unified virt verbs interface for both RDMA APP

Main contributions:

* Design
* Implementation and evaluation

1. Background

2.1 Traditional network virtualization

2.2 RDMA network

1. Overall Design

The goal of uniRDMA is to provide an unified vRNIC for each VM or container, and applications can use RDMA via uniVerbs interface for a virtual network on top of the virtual interface in an unmodified way. Ideally, the performance of the virtual network should be close to bare-metal RDMA, and policies on both control and data path are flexible to be configured purely in software.

In this section, we present the system architecture and key challenges in the design of FreeFlow.

3.1 Rational: RDMA is separate control and usage by RDMA resources

* Control: create and map resource
* Usage: direct use resource

3.2 Overall Design:

* Virtual layer: create vRNIC? build and map resource
* Virt verbs: use vRNIC

1. Virtual Layer

4.1 Space choice

4.2 vRNIC virtualization

4.3 vRNIC mapping

4.4 virtual RDMA network

1. Virt Verbs
   1. basic interface
   2. interface optimization
2. Evaluation

6.1 Methodology

6.2 Basic benchmark: latency, throughout, scalability

6.3 Real-world Applications: Graph-500，Spark-RDMA

1. Discussion

Security: shared memory, mapping DB by PFN

Work with native RDMA: support work in one host, support communicate

TCP/IP on RDMA

Live migration

Congestion control

1. Conclusion