# Performance of the 25Gbps/100Gbps fullmesh RoCE network using Mellanox ConnetX-4 Lx adapter and Ruijie S6500 Ethernet switch [1]

Hui Li, Xiaolong Chen,Tao Song, Haiyin Chen, Hao Chen
Testing Group of the Data Center Network at Ruijie Network
*Ruijie Network Fuzhou, China*
{huili, chenxiaolong, songtao, chy, chenh}@ruijie.com.cn

*Abstract*— **Zero-copy RDMA is promising in the support of IPC operations for a wide range of applications. Modern high-performance network protocols such as Infiniband, RoCE, or iWARP offer RDMA, which provides high bandwidth and low latency communication for applications. In this paper, we assess 25Gbps/100Gpbs fullmesh RoCE network using the Mellanox ConnetX-4 Lx adapter and Ruijie s6500 switch, which are already largely deployed in Alibaba, Baidu, and Tencent data center network environments. We find that the capability of the switch can be a bottleneck in the fullmesh network, and we also find that our fullmesh deployment can achieve linear scalability of throughput as long as the total throughput of servers does not exceed the link rate of switches.**

***Keywords: RDMA, RoCE, Network, Fullmesh***

## I. INTRODUCTION

Computation, storage, and the web are the three main types of applications deployed in data center or cloud. All these applications require IPC utilities to exchange intermediate data or control messages to process parallel tasks that cross over many servers. As such, the overhead of moving data within the data center cannot be overlooked. Remote direct memory access (RDMA) provides the ability of a device to access remote host memory directly, without the intervention of the computer's operating system. The RDMA enables high-throughput, low-latency networking with low CPU utilization, which is especially useful in parallel compute clusters.

The RDMA has the following major advantages: 1) Zero-copy, applications can perform data transfer without the network software stack involvement and the data are sent and received directly to the buffers without being copied between the network layers. 2) Kernel bypass, applications that can perform data transfer directly from the user space without the need to perform context switches. 3) No CPU involvement, applications can access remote memory without consuming any CPU in the remote machine. The caches in the remote CPU(s) will not be filled with the accessed memory content.

There are several network protocols that support RDMA: 1) InfiniBand, a new generation network protocol that supports RDMA natively from the beginning. 2) RoCE, A network protocol that allows RDMA to be performed over Ethernet networks. 3) iWARP, A network protocol that allows performing RDMA over TCP. In this paper, we are deploying Remote Direct Memory Access (RDMA) technology in fullmesh RoCE network to provide ultra-low latency and high throughput network to applications, with very low CPU overhead.

InfiniBand (abbreviated IB) is a computer-network communication standard used in high-performance computing that give very high throughput and very low latency. It is used for data interconnect both among and within computers. InfiniBand is also used as either a direct or switched interconnection between servers and storage systems.

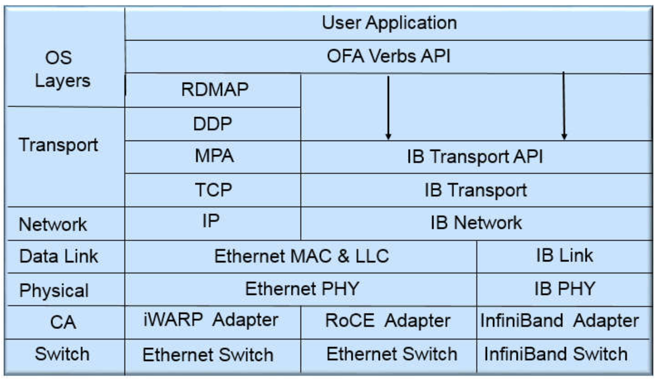

Figure 1: RDMA software stack

RoCE is the transport layer standard that provides for RDMA over an Ethernet network. Unlike the InfiniBand, with RoCE you can use the same Ethernet cabling and switching that you already have while it provides the capability of RDMA. There are two RoCE versions, RoCE v1 and RoCE v2. RoCE v1 is an Ethernet link layer protocol and hence allows communication between any two hosts in the same Ethernet broadcast domain. RoCE v2 is an internet layer protocol which means that RoCE v2 packets can be routed. As depicts in Figure 1, RoCEv2 retains the IB transport layer, but replaces IB networking layer (L3) with IP and UDP encapsulation, and replaces IB L2 with Ethernet.

iWARP is an alternative RDMA offering that is more complex and unable to achieve the same level of performance as RoCE-based solutions. iWARP uses a complex mix of layers, including DDP (direct data placement), an adaptation known as MPA (marker PDU aligned framing), and a separate RDMA protocol (RDMAP) to deliver RDMA service over TCP/IP. This convoluted architecture is an ill-conceived attempt to fit RDMA into the existing software transport framework. Unfortunately this compromise causes iWARP to fail to deliver the three key benefits that RoCE is able to achieve: high throughput, low latency, and low CPU utilization.

In network logical topology, communication devices are modeled as nodes and the connections between the devices are modeled as lines between the nodes. Several types of network logical topologies have been used in LANs, including ring, bus,

[1] WAINA-2019 The 33-rd International Conference on Advanced Information Networking and Applications at Kunibiki Messe Matsue, Japan

mesh and star. Figure 2 depicts a fullmesh with 11 nodes, which is a fully connected graph. In figure 4, the nodes were connected to the switch in a bus topology. We also consider Figure 4 as a fullmesh network in logically. In Figure 4, any two nodes are connected through only 1 hopper or 2 hopper switch devices. As a result, the bandwidth and the latency between any two nodes of a fullmesh network are optimized.

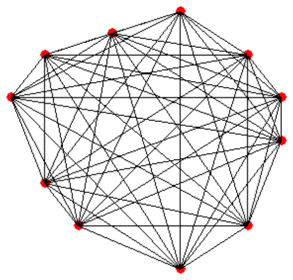
Figure 2: Fullmesh with 11 nodes

This paper uses OFED, MPI, and Spark to evaluate the performance of 25Gbps/100Gbps fullmesh RoCE network. OFED refers to OpenFabrics Enterprise Distribution. It is a package for Linux that includes all of the needed software (libraries, utilities and drivers) to work with RDMA enabled protocols (Infiniband, RoCE, and iWARP). It includes a PerfTest library that we will use to evaluate the performance of 25Gpbs/100Gpbs fullmesh RoCE network. The message passing interface (MPI) is a standardized and portable message-passing standard designed by a group of researchers from academia and industry. There are several MPI runtime implementations, which include Intel MPI, MPICH, and OpenMPI. All three MPI implementations support the RDMA protocol. We used the OpenMPI MPI_Alltoall API to evaluate the performance of 25Gbps/100Gpbs fullmesh RoCE network. Spark is a unified analytics engine for large-scale data processing. Spark powers a stack of libraries including SQL, DataFrames, and MLlib for machine learning. Mellanox sponsors the SparkRDMA project which have Spark to use RDMA protocol with only several extra configuration work of Spark. In this paper, we also use the SparkRDMA Terasort application to evaluate the performance of 25Gpbs/100Gpbs fullmesh RoCE network.

The rest of the paper is organized as follows. We provide a brief overview of the related work in Section 2. We illustrate the deployment architecture of the fullmesh network in Section 3. In Section 4, we introduce four applications and evaluate the performance of 25Gbps/100Gpbs fullmesh RoCE network. Finally, we conclude the paper in Section 5.

## II. RELATED WORK

### *A. Next generation fullmesh network*

Paper[1] evaluated the performance of the RDMA over RoCE standard in an enterprise data centers infrastructure. It also analyzed why the RoCE has received broad industry support from many hardware, software and system vendors. The authors in paper[2] discussed the latest development in RoCE congestion control. Hardware congestion control reduces the latency of the congestion control loop; the loop reacts promptly to congestion by throttling the transmission rate quickly and accurately. The authors also surveyed architectural features that allow deployment of RoCE over lossy networks and presented test results. Paper[3] designed a DSCP-based priority flow control (PFC) mechanism to ensure large-scale deployment of RoCEv2. They addressed the safety challenges brought by PFC-induced deadlock, RDMA transport livelock, and the NIC PFC pause frame storm problem. Their experiments showed that the safety and scalability issues of running RoCEv2 at scale can be addressed. Paper[4] describes the challenges of using RDMA over commodity Ethernet (RoCEv2) to support some of Microsoft's highly-reliable, latency-sensitive services. Their experiences show that the safety and scalability issues of running RoCEv2 at scale can all be addressed, and RMDA can replace TCP for intra data center communications and achieve low latency, low CPU overhead and high throughput.

### *B. Parallel runtimes using RDMA*

Paper[5] focused on the redesign of the communication framework inside MapReduce. They proposed the design of an RDMA-based Hadoop MapReduce over InfiniBand and several design elements, such as data shuffle over InfiniBand. Their results show that the RDMA-based solution achieves a performance improvement of 32% over IP-over InfiniBand (IPoIB) and 21% over Hadoop-A. Paper[6] presented a novel design of HDFS using RDMA over Infiniband via JNI interfaces. Experimental results showed that, for 5GB HDFS file writes, the new design reduces the communication time by 87% and 30% over 1Gigabit Ethernet and IP-over-Infiniband, respectively, on the QDR platform. Paper[7] focused on how RDMA and high-performance interconnects can benefit Spark. It utilized the plug-in based approach with the staged event-driven architecture and RDMA-based designs provided both performance and productivity. The performance results showed that Spark applications can benefit from an RDMA enhanced design. Paper [8] explored the potential of employing RDMA to improve the performance of online data processing (OLDP) workloads on MySQL using Memcached for real-world web applications. This outcome revealed that RDMA-enhanced Memcached can significantly improve the performance of OLDP workloads that perform a sizeable number of concurrent reads compared to default Memcached running with IPoIB QDR. Paper[17] rethink the widely used RPC abstraction for network communication because the deep larning workloads and network technologies such as RDMA are emerging. By designing a simple "device"-like interface, along with a combination of static analysis and dynamic tracing, they have enabled cross-stack optimizations for general deep neural network training to take full advantage of the underlying RDMA capabilities.

### C. *Main contribution of this paper*

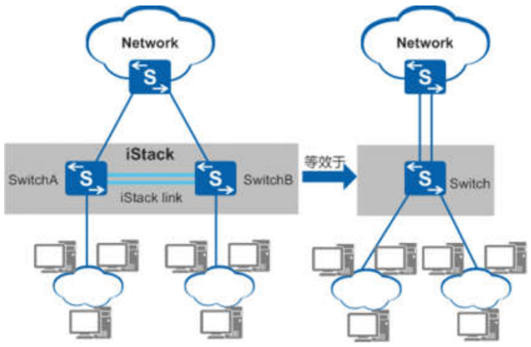

Figure 3: Intelligent Stack

One of the main contributions of this paper is to make a thorough performance evaluation of the 25Gbps/100Gbps fullmesh RoCE network using state-of-art hardware devices. We also note several performance issues when using the fullmesh of RoCE network for four state-of-the-art applications. The second contribution of this paper is addressing how difficult it is to adapt RDMA to fullmesh applications. We show that it does not require much programming effort to adapt RDMA to fullmesh applications.

## III. Deployment Architecture of 25Gbps/100Gbps Fullmesh RoCE Network

Intelligent stack [9] (iStack) connects multiple switch devices into one switch device. iStack is a nonstandard technology in DCN that is used to extend the ports of a switch. Since it can simplify network management, iStack has been widely used in today's DCN network deployment. However, iStack requires that multiple switch devices must come from the same vendor, and be installed with the same software, which conflicts with the openness of the next generation DCN network. Nevertheless, it is easy to dismantle the iStack if there is a fault between switch devices. The fault of one switch easily affects the work of other switches.

VSU lite [10] technology can overcome the disadvantages of iStack while still providing the advantages of iStack. VSU lite utilizes dual ports to uplink two switch devices, and NIC is set as bond mode 4, which obeys the IEEE 802.3ad (RoCE LAG). To make use of VSU lite, it requires modification of the network driver in the Linux kernel in addition to some configurations in the switches. Figure 4 illustrates the deployment architecture of 25Gbps/100Gpbs fullmesh RoCE network using VSU lite. In Figure 4, there are two groups of servers connected to 4 ASWs. The intragroup message go through ASW, while the intergroup message go through ASW, PSW.

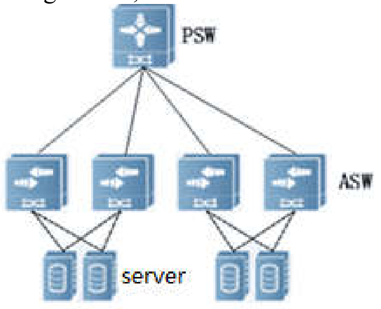

Figure 4: VSU lite

Priority flow control (PFC) is a recommended flow control mechanism that allows for high RoCE performance and guarantees zero packet loss in the network. Initial implementations of RoCE required a lossless network for optimal performance. This is typically achieved by enabling PFC on Ethernet NICs and switches. Lossless networks avoid dropping packets when the buffer is full by using PFC to pause the incoming packet transmission before the buffers that receive the packets overfill. PFC complements Congestion Notification in Data Center Bridging networks. Operation of priority-based flow control is limited to a domain controlled by a Data Center Bridging control protocol that controls the application of Priority-based Flow Control, Enhanced Transmission Selection, and Congestion Notification.

For the best performance, we configure the PFC, ECN, DSCP, DCQCN services in network adapters and switches for the RoCE network. Explicit congestion notification (ECN) was initially defined for TCP/IP by embedding an indication for congestion in the IP header and an acknowledgement in the TCP header. ECN compatible switches and routers mark packets when congestion is detected. The congestion indication in IP header is also used by congestion control of RoCEv2. Differentiated services code point (DSCP) is an 8-bit field in the IP header. The network element will take the priority from the DSCP bits and map it to the right buffer/queue/priority. DC-QCN is a congestion control protocol for RoCE. It is based on the ECN feature on network switches, to inform the sender to throttle the injection rate upon congestion.

Our testbed has16 Dell R720 servers. Each compute node has a 16-core 2.4 GHz Intel Xeon E5-2630 processor with 16 GB of main memory, and it is equipped with Mellanox ConnectX-4 LX (dual port 25GbE/50GbE) HCAs. We use CentOS Linux 7.4.1160 (Core) with kernel 3.10.0-693.2.2.el7.x86_64 with VSU lite patch. In addition, we use the Mellanox Open Fabrics Enterprise Distribution (OFED) [11] MLNX_OFED_LINUX-4.2-1.0.0.

ConnectX-4 Lx EN is the adapter supports RoCE network produced by Mellanox. It provides an unmatched combination of 1, 10, 25, 40, and 50 GbE bandwidth, submicro second latency and 75 million packets per second message rate.

The RG-S6500 is an Ethernet switch produced by the Ruijie network. It is a high-performance and high-density switch designed for a DCN network. It provides up to 48 25Gbps ports and 8 100Gbps ports. It has four 1GHz CPUs, 2GB RAM, and 4x4 MB cache.

## IV. Performance of 25Gpbs/100Gpbs Fullmesh RoCE Network

### A. *performance of Point to Point using MPI PingPong*

We make two sets of experiments to evaluate performance of RDMA of point to point operations by using Intel's MPI PingPong jobs [12]. The first set of experiment is the s-s pattern, which means that the server connects to the server directly with no switch. The second set of experiment is the s-sw-s pattern, which means that two servers are connected through a switch.

```
Function fullmeshserver(){
for h  in $hosts
do
   for p in $ports
      do
             ssh $h ib_write_bw –p $p –q 200 –D 1000 –tclasss=136 > /dev/null 2>&1 &
      done
done
}

Function fullmeshclient(){
Step1 = 0
Step2  =0
Size = 10 #//the size of all nodes
For h1 in $hosts
Do
For h2 in $hosts
   Do
      If [ “$h1”x != “$h2”x ]; then
         If [ $step1 –lt $(size-1)] ]; then
         Ssh $h1 ib_write_bw –p ${ports2arr[$step1]} –q 200 –D 1000 –tclass=136 $h2 >/dev/null 2>&1 &
         Fi
         If [ $step1 –eq $(($size-1)) ]; then
         Ssh $h1 ib_write_bw –p ${ports2arr[$step2]} –q 200 –D 1000 –tclass=136 $h2 >/dev/null 2>&1 &
Step2 = $(($step2+1))
Fi
Fi
Done
Step1 = $(($step1+1))
Done
}
```

Figure 6: Pseudo code of OFED fullmesh scripts

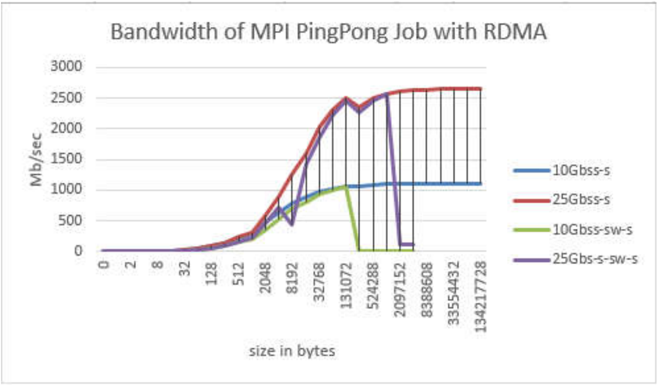

Figure 5: Bandwidth of Intel MPI PingPong job

In the two set of experiments, the PingPong test sends and receives small message for 1000 times using MPI sends/recv primitives. Figure 5 shows that the 25Gbps s-s pattern and 10Gpbs s-s pattern can provide stable throughput performance for message sizes varying from 1 byte to 134MB, while the 25Gpbs s-sw-s pattern and 10Gpbs s-sw-s pattern have very serious performance degradation when message sizes are larger than 5KB. There are several reasons for the performance degradation. The first reason is network congestion caused by the buffer overflow of the RG-S6500-4c switch. In other word, the RG-S6500-4c switch does not have sufficient buffer to transfer the data sent out by Mellanox ConnectX-4 Lx adapter. Second, we did not configure the priority flow control (PFC) for this experiments, which results in a significant performance penalty for network congestion.

### *B. performance of Fullmesh using OFED ib_write_bw*

We made the performance experiments of 25Gpbs/100Gpbs fullmesh RoCE network using OFED scripts with up to 16 physical servers. In this experiment and all the experiments following this one, the topology of experiments is shown in Figure 4. There were two groups of servers, each of which had 8 physical servers. In each group, the physical servers were connected to 2 dedicated ASW switches with VSU lite enabled. Messages between two groups had to go through PSW as shown in Figure 4. We use 8 servers to mimic single-rack traffic, and use 16 servers to mimic inter-rack traffic. Figure 6 illustrates the pseudo code of the fullmesh of OFED scripts for both the client and server functions. It has only 30 code lines. In the pseudo code, we spawn N-1 server processes and N-1 client processes on each server to mimic the fullmesh communications of N*(N-1) connections. We also set the number of queue pairs to 200 to increase bandwidth utilization for data transferring. In addition, there is only little performance fluctuations in bandwidth when setting up the queue pairs as 200.

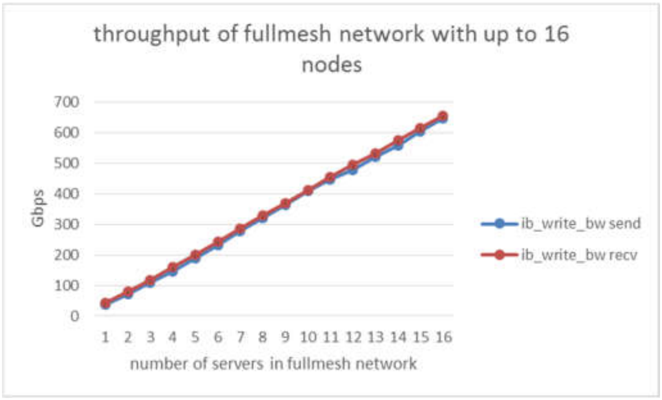

Figure 7: Throughput of fullmesh network using OFED scripts with up to 16 nodes

Figure 7 shows the accumulated throughput of send and receive of OFED ib_write_bw scripts using up to 16 physical servers. The bandwidth of single server is more than 45Gbps, and this is because we use RoCE LAG technology which leverages the full capability of ConnectX-4 Lx adapter. The throughput is shown to be stable and linearly scalable, the throughput is linearly scalable with the number of physical servers. The results indicate that the total throughput over 16 physical servers , which is 650Gpbs, did not exceed the link rate of ASW and PSW. As a result we do not see the performance bottleneck of throughput of fullmesh network with the 16 nodes.

We should note that it may show the performance bottleneck of fullmesh network when the total throughput of all nodes is more

```
Int main(int argc, char *argv[]){

Int rank, size;
MPI_Init(&argc,&argv);
MPI_Comm_size(MPI_COMM_WORLD,&size);
MPI_Comm_rank(MPI_COMM_WORLD,&rank);

Int sendcount = 1024*1024;
Int recvcount = sendcount;
Printf("size:%drank:%d
            sendcount:%d\n",size,rank,recvcount)
Int len = sendcount*size;
Int *sendbuf = new int[len];
Int *recvbuf = new int[len];
If(rank == 0)
            Printf("MPI_Alltoall repeat: 1000\n");
For(int I = 0;i<1000;i++)
            MPI_Alltoall(sendbuf,sendcount,MPI_INT,recv
buf,recvcount,MPI_INT,MPI_COMM_WORLD);
MPI_Barrier(MPI_COMM_WORLD);
Delete [] sendbuf;
Delete [] recvbuf;
MPI_Finalize();
Return 0;

}
```

Figure8: Pseudo code of invoking MPI_Alltoall

than 800Gbps, which is the link rate of PSW in Figure 4.

## C. *performance of Fullemsh using MPI Alltoall*

MPI_Alltoall is an MPI collective operation [13] in which all processes send the same amount of data to each other, and receive the same amount of data from each other. Each process breaks up its local sendbuf into n blocks. Process j sends the k-th block of its local sendbuf to process k. The amount of data sent must be equal to the amount of data received.

Figure 8 is the pseudo code of invoking MPI_Alltoall. The code is very simple, and it has only 21 code lines. We set sendcount as 1024*1024 to have high network throughput. We set the repeat number to 1000 to make the MPI_Alltoall primitives run 1000 times so as to have sustainable high network throughput.

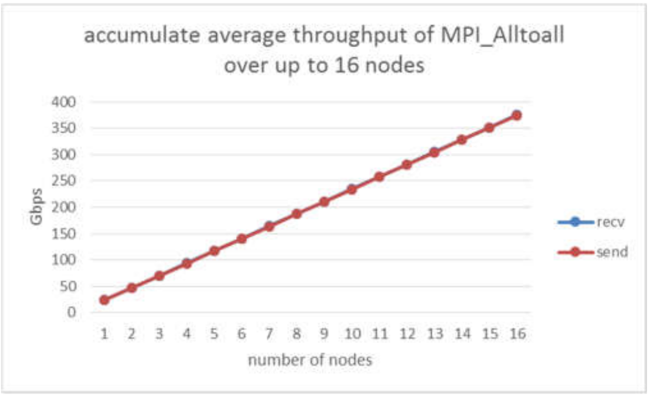

Figure 9: Throughput of fullmesh network using MPI_Alltoall over up to 16 nodes.

Figure 9 shows the total throughput of fullmesh network using MPI_alltoall over up to 16 nodes. The throughput is shown to be stable. It also shows linear scalability of throughput using up to 16 nodes. The average throughput of each server node is nearly 25 Gbps, which is approximately half the 50Gbps, the link rate of ConnectX-4Lx adapter. The reason is that we use the rdma_cm lib of OFED for this MPI program, which does not use the capability of the RoCE LAG. As a result, it only leverages half capability of ConnectX-4 Lx adapter.

## D. *performance of Fullmesh using Spark Terasort*

Terasort [14][16] is an important benchmark that measures the amount of time to sort one terabyte of randomly distributed data on a given computer system. SparkRDMA Terasort is the Terabyte sort program run on SparkRDMA [15]. We have 3 Spark applications to run the SparkRDMA Terasort applications:

1. TeraGen is a map/reduce program to generate the data
2. TeraSort samples the input data and uses map/reduce to sort the data into a total order.
3. TeraValidate is a map/reduce program that validates the output is sorted.

Terasort is a standard map/reduce tasks, except a custom partitioner. The custom partitioner uses a sorted list of N-1 sampled keys that define the key range for each reduce. In particular, all keys such that sample[i-1]<=key<sample[i] are sent to reduce i. It guarantees that the output of reduce i are all less than the output of reduce i+1. To speed up the partitioning, the partitioner builds a two level trie that quickly indexes into the list of sample keys based on the first two bytes of the key. TeraSort generates the sample keys by sampling the input before the job is submitted and writing the list of keys into HDFS.

We ran the Terasort code on 16 nodes and sorted the 100 million records (10GB) within 5 minutes. Figure 10 show the total throughput of fullmesh network using Spark Terasort with up to 16 nodes. The throughput is shown to be stable. It also shows the linear scalability of the total throughput using up to 16 server nodes. However, we see slightly fluctuations of throughput when number of nodes are 2,3,9, and 13. This is because the input data is a little skewed so we could see the unfair throughput. The result of the average throughput of each node is nearly 300Mbps; this value is very small because the disk I/O operations are involved in the shuffle stage of the Spark Terasort task. In fact, even considering the peak performance of throughput of Terasort Spark task, which is about 2Gbps, and the total peak throughput is 2*16 = 32Gpbs, which is still much lower than the link rate of the switch.

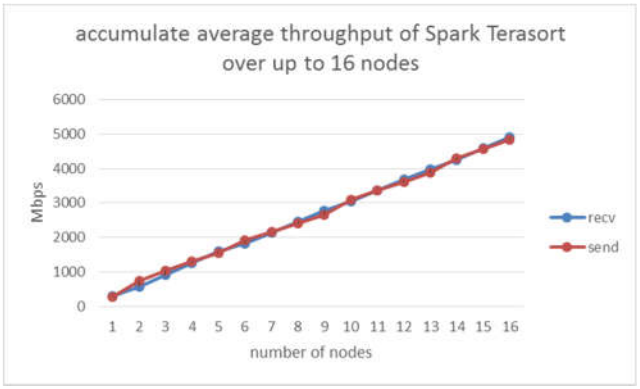

Figure 10: Throughput of fullmesh network using Spark Terasort over up to 16 nodes.

## V. Summary and Conclusion

This paper depicts the deployment architecture of 25Gpbs/100Gpbs fullmesh RoCE network using VSU-lite. This paper studies the performance of throughput of 25Gpbs/100Gpbs fullmesh RoCE network using four applications, the MPI PingPong jobs, the OFED scripts, MPI Alltoall primitives, and SparkRDMA Terasort. The results show that the capability of switch and performance of disk I/O could be a performance bottleneck in the fullmesh network. This paper also shows linear scalability of total throughput of fullmesh network using up to 16 nodes as long as the total throughput does not exceed the link rate of switch. We believe that our deployment architecture and configurations of 25Gpbs/100Gpbs fullmesh RoCE network are optimal and can be widely used in academia and industrial environment.

**Acknowledgements**

We highly appreciate the kind advice provided by the technical support team at Mellanox in China, they answered many of our questions, and provided some excellent suggestions regarding setting up the experiments. This paper is support by the project of OS1.0PS1(ALI) in Ruijie which is initiated by the cooperation with Alibaba group in 2018.

## VI. References

[1] Motti Beck, Michael Kagan, Performance Evaluation of the RDMA over Ethernet (RoCE) Standard in Enterprise Data Centers Infrastructure.Proceedings of DC-CaVES2011

[2] Chuanxiong Guo, Haitao Wu, Zhong Deng, Gaurav Soni, Jianxi Ye, Jitu Padhye,Marina Lipshteyn, RDMA over Commodity Ethernet at Scale.SIGCOMM'16

[3] Alexander Shpiner, Eitan Zahavi, Omar Dahley, Aviv Barnea, Rotem Damsker, Gennady Yekelis, Michael Zus, Eitan Kuta and Dean Baram, Yokneam, RoCE Rocks without PFC: Detailed Evaluation

[4] Yibo Zhu et al. Congestion Control for Large-Scale RDMA deployments. In ACM SIGCOMM, 2015

[5] N.S.Islam, M.W.Rahman, J.Jose, R.Rajachandrasekar, H.Wang, H.Subgramoni, C.Murthy & D.K. Panda, High-Performance RDMA-based Design of Hadoop MapReduce over InfiniBand. SC '12 Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis

[6] X Lu, NS Islam, M Wasi-Ur-Rahman, J Jose, H Subramoni, High-Performance Design of Hadoop RPC with RDMA over InfiniBand.International Conference on Parallel Processing, 2013

[7] Accelerating Spark with RDMA for Big Data Processing: Early Experiences

[8] Dipti Shankar, Xiaoyi Lu, Jithin Jose, Md. Wasi-ur-Rahman, Nusrat Islam, Dhabaleswar K. Panda "Can RDMA benefit online data processing workloads on memcached and MySQL". ISPASS, Philadelphia, PA,USA 2015

[9] http://support.huawei.com/enterprise/docinforeader!loadDocument1.action?contentId=DOC1000074401&partNo=10042 Stack Configuration – Huawei Technical Support

[10] http://www.ruijie.com.cn/fa/xw-hlw/61546 How to implement the VSU-lite in DCN network

[11] https://www.mellanox.com/related-docs/prod_software/Mellanox_OFED_Linux_User_Manual_v4.0.pdf Mellanox OFED for Linux User Manual

[12] https://software.intel.com/en-us/articles/intel-mpi-benchmarks

[13] https://www.open-mpi.org/doc/v1.10/man3/MPI_Alltoall.3.php

[14] https://github.com/ehiggs/spark-terasort

[15] https://github.com/Mellanox/SparkRDMA

[16] TeraByte Sort on Apache Hadoop, Owen O'Malley Yahoo! May 2008

[17] Jilong Xue, Youshan Miao, Cheng Chen, Ming Wu, Lintao Zhang, Lidong Zhou, RPC Considered Harmful: Fast Distributed Deep Learning on RDMA, Microsoft Research 2018.