Comparing Ethernet and Soft RoCE for MPI Communication

Gurkirat Kaur[1], Manoj Kumar[2], Manju Bala[1]
[1]*Department of Computer Science & Engineering, CTIEMT Jalandhar, Punjab, India*
[2]*Department of Electronics and Communication Engineering, CTIEMT, Jalandhar, Punjab, India*

Abstract: *RDMA has pleasant consideration in the late 1990 when the Virtual Interface Architecture was introduced. This growth has accelerated with the introduction of Open Fabrics Alliance's (OFA's) Verb Interface. The stability & independence of OFA verb interface facilitated significant growth of software applications that exploit the benefits of RDMA. RoCE is an emerging trend that can be made to work on the Ethernet infrastructures. In this paper, we evaluate the Linux cluster, having multi nodes with fast interconnects i.e. Gigabit Ethernet & Soft RoCE and evaluates its performance using IMB and OSU Micro Benchmarks. A comparison between the IMB and OSU Micro benchmark is done and our results shows that IMB Benchmark is performing better in case of collective Benchmark class as OSU Micro benchmark is performing better in other classes of benchmark.*

Keywords: *Ethernet, RDMA, RoCE, Soft RoCE*

I. Introduction

In terms of HPC interconnects, there are several network interconnects that provide ultra-low latency (less than 1microsecond) and high bandwidth (several gigabytes per second). Some of these interconnects may provide flexibility by permitting user-level access to the network interface cards for performing communication, and also supporting access to remote processes' memory address spaces. Examples of these interconnects are Myrinet from Myricom, Quadrics and InfiniBand [1]. The focus of our paper is on the RoCE (RDMA over Converged Ethernet) which allow the users to take the advantage of low latency, high efficiency; high performance. RoCE is basically an InfiniBand (IB) protocol that can be used over the Ethernet infrastructures. RoCE provide all of the InfiniBand transport benefits and well established RDMA ecosystem combined with converged Ethernet. RoCE is network protocol which allows RDMA access over the Ethernet. It is also called link layer protocol which allows the communication between the two hosts on the same Ethernet broadcast domain [3]. RoCE uses a RDMA technology that helps in reducing the system load and also improves the throughput. Many Linux Distributors included OFED (Open Fabrics Enterprise Distributors), support a wide and rich range of middle wares and application solutions like IPC, sockets, messaging, virtualization etc. RoCE is implemented & available at the OFED stack. RoCE can be implemented in Hardware as well as software. April 22, 2010 – System Fabric Works (SFW) is a systems integration company delivering a high quality integration, development & deployment of high performance software solutions to the global clients. SFW is delivering powerful, open-source fabric and I/O solutions in high performance software engineering, announced support for RDMA over Converged Ethernet (RoCE) implemented in software as an addition to the Open Fabrics Enterprise Distribution (OFED) release 1.5.1 for Linux. RoCE is a new standard announced earlier by the InfiniBand Trade Association (IBTA) and supported by the Open Fabrics Alliance. SFW is announcing the availability of a software implementation of the RoCE standard – compatible with standard Ethernet networks – called "**Soft RoCE**." With **Soft RoCE**, SFW offers the opportunity for data center technologists to implement RDMA for their business solutions to improve computing efficiency, simplify infrastructure, and future proof their networks for scaling from 1 to 10 gigabits per second. [4]

Our objective of this paper is to evaluate the performance of Linux cluster using two of the most commonly used MPI implementations. We first build a Linux Cluster subsequently, we evaluated the performance of the two of the most commonly used MPI implementations in the HPC industry, which are OSU Micro Benchmark which is based on the MVAPICH2 and Intel MPI using the Intel MPI Benchmark utility (IMB).

The rest of the paper is organised as follows: In Section 2, we present our cluster design. In section 3, we give the overview of MPI Implementations. In section 4, we evaluate our experimental results and interpret the benchmark results. We state our conclusion in the last section.

II. MPI Implementations

In this section, we perform the Intel MPI and OSU Micro benchmarks for our Linux cluster. We used Intel's MPI Benchmark and Ohio State University (OSU) Micro Benchmark to measure and compare the performance of the two interconnect i.e. Ethernet and Soft RoCE of the Linux cluster.

1.1. Intel's MPI Benchmark

Intel MPI is a multi fabric message passing library that is based on message passing interface, v2 (MPI-2) specifications, Intel MPI library focus on making the application to perform better on the Intel Architecture based cluster. This MPI implementation enables the developers to upgrade or to change the processors & interconnects as new technology become available without doing changes in the software or the operating system environment. This benchmark provides an efficient way to measure the performance of a cluster, including node performance, network latency and throughput.IMB 3.2.4 is categorized into 3 parts: IMB-MPI1, IMB-EXT, and IMB-IO. We will focus on the IMB-MPI1 which is used in our evaluation. The IMB-MPI1 benchmarks are classified into 3 classes of benchmarks: Single Transfer Benchmark & Collective Transfer Benchmark.

1.2. OSU Micro Benchmark

The Ohio Micro Benchmark suite is a collection of independent MPI message passing performance micro benchmarks developed and written at The Ohio State University. It includes traditional benchmarks and performance measures such as latency, bandwidth and host overhead and can be used for both traditional and GPU-enhanced nodes. It is a suite of micro-benchmarks for testing various MVAPICH2 MPI operations. The OSU Micro Benchmark (OMB) suite has been the most widely used set of benchmarks to compare the performance of different MPI libraries on clusters. We will focus to measures the point to point MPI Benchmarks and Collective Benchmarks using OSU Micro Benchmark. OSU benchmark is divided into 2 categories: Point to Point MPI Benchmark and Collective Benchmark.

Single Transfer and Point to Point Benchmark are used to measure the bandwidth and latency tests. Collective Benchmarks measure the time needed to communicate between a group of processes in different behaviours. There are several benchmarks of this category and the following is description of the collective benchmarks that was used in our evaluation:

- **Reduce Test:** In Reduce benchmark each process sends a number to the root & then total number will be calculated by the root.
- **AlltoAll Test:** In MPI_AlltoAll routine, all processes send a message of a size equal to the chosen size * number of processes to all processes. This is an extension to MPI_Allgather where each process sends data to each receiver.
- **Gather Test:** This routine, gathers together the values from a group of processes to the root process. In this operation the number of data items collected from processes, and the data items are arranged contiguously in order of process rank.
- **Allgather Test:** MPI_Allgather routine gathers the values from all the processes together and distributes it to all the processes. This routine can be thought of as an MPI_Gather where all processes, not just the root, receive the result.
- **Reduce_scatter Test:** Perform reduction operation on vector element across all the processes, then distributes the segment of result vector to all the processes.

III. Cluster Design

To perform the Benchmarks evaluation, a setup required to be designed. This setup consists of a the heterogeneous Linux cluster design consists of 2 nodes having Intel's i5 core 2.67 GHz and Intel's i3 core 2.13 GHz processors . The Operating system running on both the Nodes are SUSE's Linux Operating System i.e. SLES 11 SP 1 with kernel version 2.6.32.12-0.7 (x86_64). Each node is equipped with a PCIe network adapter with the connection speed of up to 1 Gigabit. The MTU used for is 1500 bytes. OFED's Soft RoCE Distribution version 1.5.2 (System Fabrics Works (SFW) offers a new mechanism in its OFED release of supporting RDMA over Ethernet). We have used Intel's MPI Benchmark and OSU Micro Benchmark to run the various experiments. Secondly, a comparative analysis of Intel's MPI Benchmark and OSU Micro Benchmark are done using the Soft RoCE & Ethernet Interfaces. The results are the average of the ten test runs for all cases. To provide more close by look at the communication behaviour of the two MPI Implementations, we have used a set of micro benchmarks. They have included a basic set of performance metrics like latency, bandwidth, host overhead and throughput. The results are the average of the ten test runs for all cases.

IV. Results And Discussions

In this section, we have discussed and interpret the results obtained by the IMB and OSU benchmarks on the Ethernet and Soft RoCE Interface. This section is further divided into two sections i.e. Ethernet Interface and Soft RoCE Interface.

1.1. Ethernet Interface

Ethernet Interface can be defined as a local area network (LAN) architecture that supports data transfer at varying speeds. Using the Ethernet Interface we did the performance comparison of Ohio State University (OSU) Micro Benchmark and Intel's MPI Benchmark (IMB). We have used the Reduce Test, Latency Test, Bandwidth Test, and Gather Test to evaluate the performance of IMB and OSU Benchmark on Ethernet Interface.

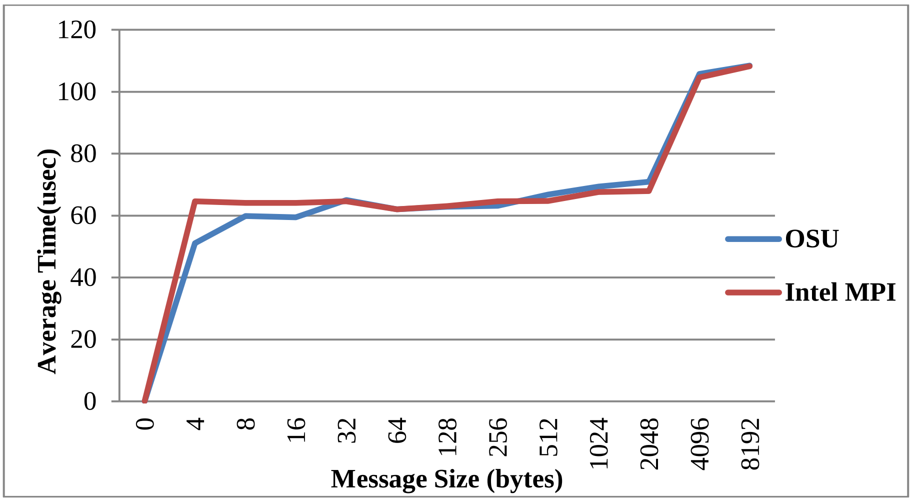

Figure 1: Ethernet_Reduce Test

In Figure 1, we used Reduce Test and it comes under Collective Benchmark category. In Reduce Test, each process sends a number to the root then the total number is calculated by the root. Here, OSU is performing better for small message size upto 128 bytes afterwards IMB's performing slight better than OSU for larger message size > 512 bytes.

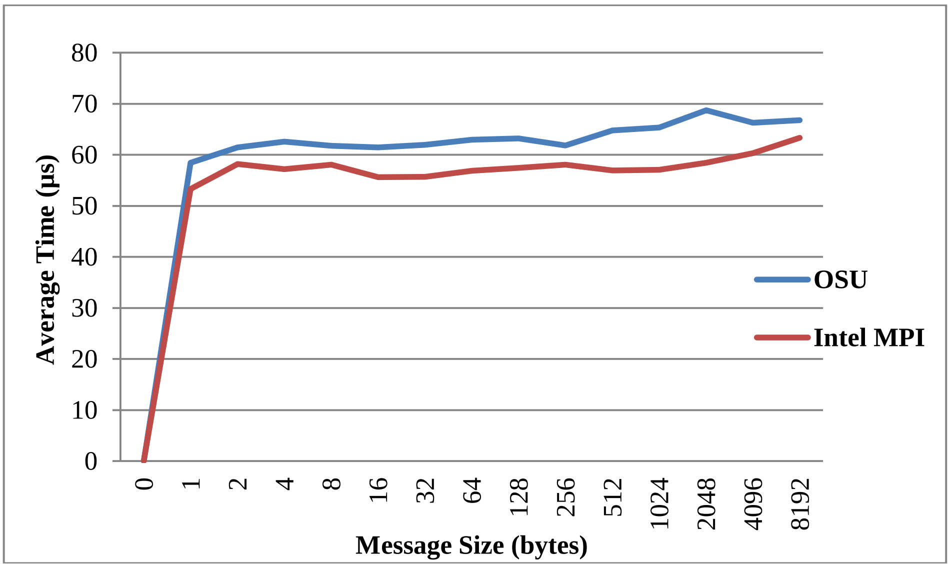

Figure 2: Ethernet_Gather Test

In Figure 2, we used Gather Test and in this test, all the processes send the same message to the root process. It is observed that Intel MPI performs better for the small and large message sizes.

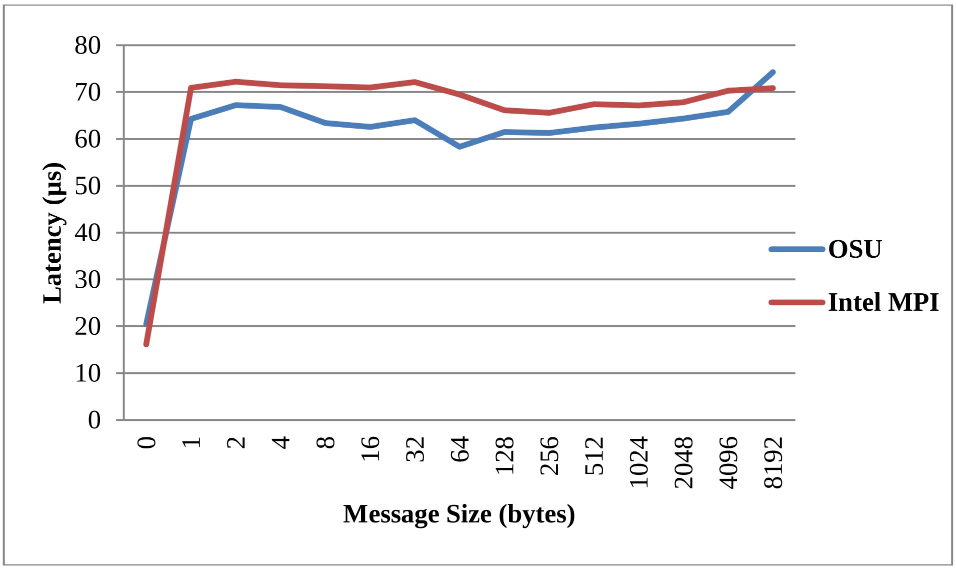

Figure 3: Ethernet_Latency Test

In Figure 3 we used Latency Test. This is done in PingPong fashion. The sender sends a message with a certain data size to the receiver and waits for a reply from the receiver. The receiver receives the message from the sender and sends back a reply with the same data size. It is observed that Intel MPI starts at good note but from message size 2 bytes the performance of OSU is increased and Intel MPI decreases.

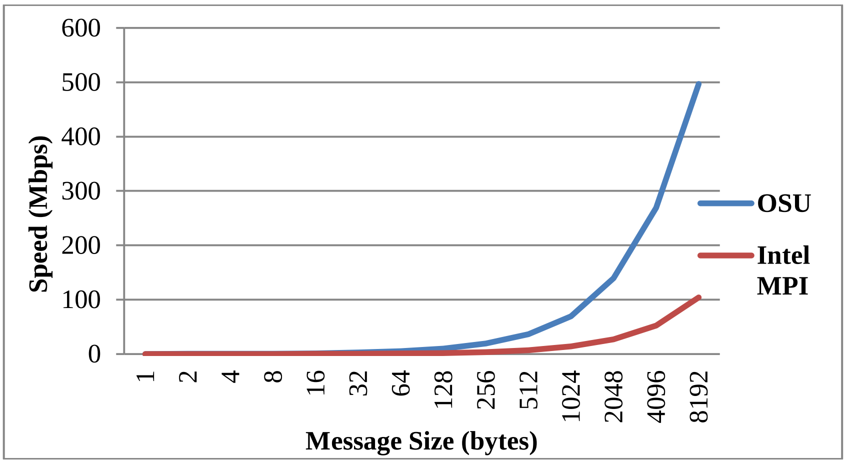

Figure 4: Ethernet_Bandwidth Test

In Figure 4, we used OSU Bandwidth test and these tests were carried out by having the sender sending out a fixed number (equal to the window size) of back-to-back messages to the receiver and then waiting for a reply from the receiver. The receiver sends the reply only after receiving all these messages. Here also, the OSU is performing better than Intel MPI for message size > 128 bytes.

1.2. Soft RoCE Interface

System Fabric Works (SFW) provides a fully software-based RoCE Linux driver called Soft-RoCE. It is an open source IB transport and network layers in software over ordinary Ethernet. It is the software implementation of hard RoCE. It interoperates with hardware RoCE at other end of wire. rxe_cfg is the configuration tool for the RXE software implementation of the RoCE protocol. To measure the performance on the Soft RoCE interface we have to start the Soft RoCE interface by using the following command:

rxe_cfg start

It will start the soft RoCE interface and afterwards we can measure the performance using the OSU and IMB Micro Benchmarks. To Stop the Soft RoCE interface the following command is used:

rxe_cfg stop

It will stop the Soft RoCE interface and the Ethernet interface will start automatically. We have used the Bandwidth Test, Latency Test, AlltoAll Test, Allgather Test, and Reduce_Scatter Test to evaluate the performance of both benchmarks on the Soft RoCE Interface..

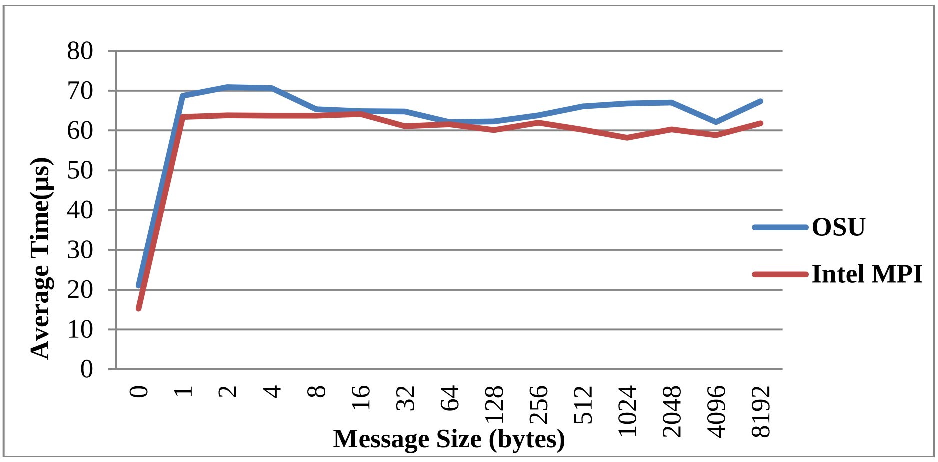

Figure 5: Soft RoCE _Latency Test

In Figure 5, we used Latency Test and it is observed that Intel MPI is performing better than OSU Benchmark. At 8k message size, the OSU Benchmark is providing 67.363µs average time and IMB is providing 61.827 µs.

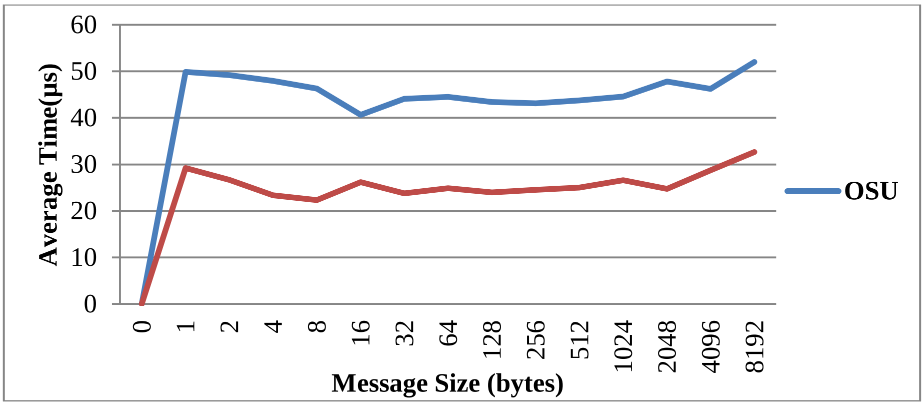

Figure 6: Soft RoCE_AlltoAll Test

As in Figure 6, AlltoAll Test, all processes send a message of a size equal to the chosen size * number of processes to all processes or all processes sends messages to all processes [19]. It is observed that Intel MPI performs better for smaller and larger message sizes.

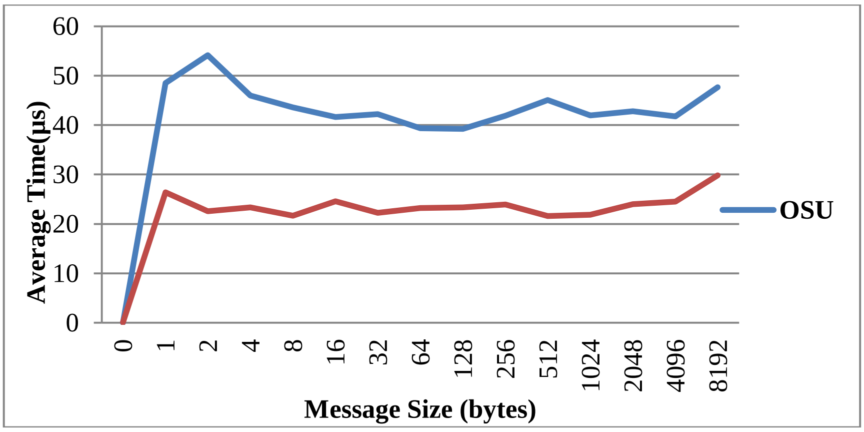

Figure 7: Soft RoCE_Allgather Test

On All Gather test, as in figure 7, every process sends X bytes and receives the gathered X*(#processes) bytes from the receivers or Allgather gathers all of the elements to all the processes [2].

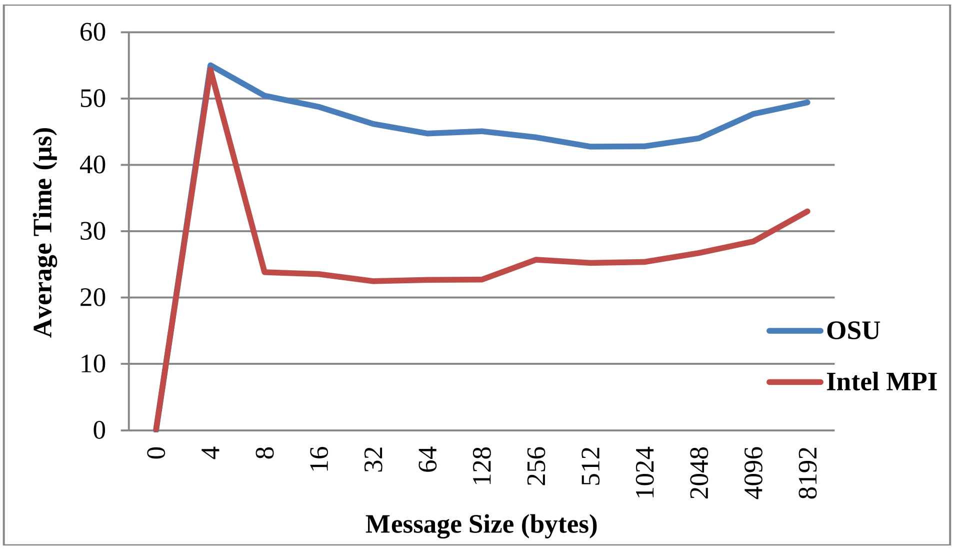

Figure 8: Soft RoCE _Reduce_scatter Test

On Reduce_scatter, as in figure 8, combines the messages from all processes at the root process and then root process scatter or broadcast the results to all processes. Here also, Intel MPI is performing almost 50% better than the OSU benchmark. Both MPI variants start at same but after 8 bytes of message size the Intel MPI starts increasing.

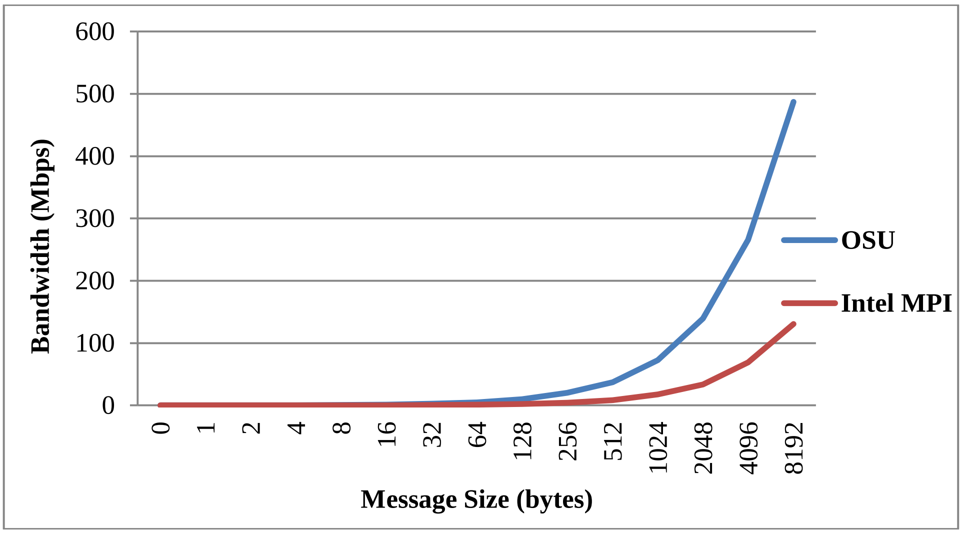

Figure 9: Soft RoCE_Bandwidth Test

Figure 9, the OSU Benchmark is performing faster than Intel MPI. For larger message size the speed for OSU is 487 Mbps and for Intel MPI the speed is 130Mbps.There is a dip in the performance of Intel MPI after the 8 bytes message size [5].

V. Conclusion

This paper presents the Linux cluster configuration & evaluates its performance using Ohio State University (OSU) Micro Benchmark and Intel MPI Benchmark (IMB) and we have done a comparison between the two MPI version micro benchmarks. It is observed that in such a cluster, OSU benchmark is performing better than Intel MPI benchmark where communication is between two processes. While Intel MPI benchmark is performing better in all other collective benchmarks where communication is between groups of processes.

References

[1] Basem Madani, raed al-Shaikh, "Performance Benchmark and MPI Evaluation Using Westmere based Infiniband HPC cluster", IJSSST, Volume 12, Number 1, page no 20-26, Feb. 2011

[2] Jalel Chergui, Isabelle Dupays, Denis Girou, Pierre-FrancoisLavallee, dimitri Lecas, Philippe Wautelet, "Message Passing Interface (MPI)", May 2013.

[3] (2013), The Wikipedia Website.[Online] available at: http://en.wikipedia.org/wiki/RDMA_over_Converged_Ethernet

[4] (2013), The System Fabrics Works Website, [Online] Available at http://www.systemfabricworks.com/news/system-fabric-works-ofed-distribution-supports-rdma- over-converged-ethernet-roce

[5] (2002-2014), The mvapich website. [Online] available at http://mvapich.cse.ohio-state.edu