Performance Evaluation of Qlogic and Mellanox QDR InfiniBand

Thomas Schoenemeyer and Hussein N. El-Harake
Swiss National Supercomputing Centre (CSCS), Manno, Switzerland
schoenemeyer@cscs.ch, hussein@cscs.ch,

Abstract – In this report we describe the performance evaluation of two QDR-InfiniBand end-to-end solutions. The evaluation is carried out on a four-node cluster using MPI micro-benchmarks provided by the Ohio State University as well as the OFED benchmark tools. Since vendors prefer certain MPI-implementations, we used both, the OpenMPI and MVAPICH implementation for the benchmarks.

Both solutions from Mellanox and Qlogic have their advantages and disadvantages. The performance differences depend upon the message size and the MPI-implementation. In some case the differences between Mellanox and Qlogic can be huge.

1. Introduction

InfiniBand as a high-bandwidth, low-latency network interconnect solution is deployed in many commodity cluster systems today. In the latest TOP500 list of June 2011, InfiniBand is used in 41.2% of all systems as the communication network (figure 1). Among the Top10 systems we have already six systems using InfiniBand as HPC communication network. We expect the rapid growth of the HPC market share to continue.

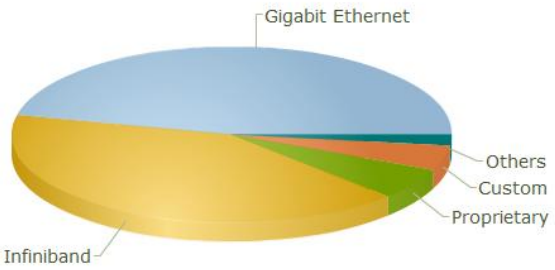

Figure 1: HPC Interconnect distribution for the TOP500 systems

After the merger of Mellanox and Voltaire [1], only two InfiniBand vendors stay in the HPC market, Mellanox and Qlogic.

InfiniBand switches and adapter cards from Mellanox and Qlogic are different in hardware design and in the associated software offered along with the hardware. Especially the approach for congestion management in order to optimize the fabric communication based on tools such as UFM or IFS [2, 3] as an important part of the end-to-end solution differs considerably.

Following the latest announcements, both companies claim to have success cases in the high-end HPC market or promote their new product roadmap. For example, QLogic recently announced to connect up to 20,000 Nodes over the next two Years for Lawrence Livermore, Sandia, and Los Alamos National Labs and claims to provide superior benchmark performance over Mellanox [4].

In June 2011 Mellanox announced their FDR products to be available at the end of the year, whereas Qlogic has not

announced FDR yet. CSCS ordered FDR hardware already for testing purposes, however in this study, only QDR technologies were compared.

Qlogic developed its own architecture called TrueScale. The TrueScale extensions to basic InfiniBand include a distributed adaptive routing capability that ensures that traffic from over-utilized links will be quickly migrated to those that are under-utilized. This algorithm is executed within each switch ASIC [5]. They also created a suite of software designed to a wide range of high-throughput workloads. Each of the software components addresses a small part of the optimization spectrum, but QLogic offers all components collectively as InfiniBand Fabric Suite (IFS) including intelligent routing [6]. It is delivered with the Qlogic switches and HCAs and it is not locked into any particular MPI library or system topology. End users can implement whichever MPI libraries or fabric topologies best suit their application workflows.

Mellanox most popular product is the QDR InfiniBand Switch IS5035, built with Mellanox's 4th generation InfiniScale® switch device, that provides up to 40Gb/s full bidirectional bandwidth per port [7]. The FabricIT™ fabric management is included. Different to Qlogic, a wide range of additional tools are available such as UFM, FCA, TARA etc. and are marketed separately. These packages need to be purchased and are not included per default. The most important tool among those is probably UFM which automatically discovers fabric resources and provides the resources to reduce congestions. It also monitors the fabric resources and traffic in real-time. The mechanisms rely on a subnet monitor to poll the switches for traffic information; the subnet monitor then gives routing instructions to the switches and assures an effective load balancing.

2. Experimental Setup

Our experiments were performed on two different configurations. We used four dual-socket nodes with Intel processors as well as the latest generation of production switches and HCAs from both vendors, Qlogic and Mellanox, along with their preferred software configurations. The four-node setup was chosen to evaluate the impact of tools to optimize MPI collective operations. However, the four-node setup is not appropriate to observe any differences. We decided to continue with the standard ping-pong tests.

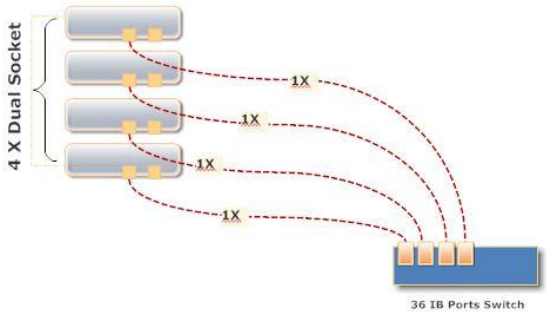

Figure 2: Setup design with 4 nodes to one QDR InfiniBand switch

Each of the four nodes deploys two Intel Xeon E5649 CPUs running at 2.53GHz and 48GB of main memory. The Operating System is SLES 11 SP1, the OFED stack is version 1.5.3.

Setup A: All nodes were connected to one Mellanox IS5035 36-port QDR InfiniBand Switch by using ConnectX3 Dual-Port QDR adapters with VPI [8].

Setup B: All nodes were connected to one managed Qlogic 12300-BS01 36-port QDR Switch linked to Qlogic 7300 HCA single-port cards with PCIe Gen2 x8 interface in each node.

As a first test we benchmarked both setups with OpenMPI 1.4.3 and MVAPICH 1.2.0. Both implementations were installed with the default values, also for the EAGER_THRESHOLD set to 12K. In this first phase we observed huge performance differences for both technologies depending on the MPI implementation. We presented the results to both vendors and asked for recommendations and their preferred setup. The conclusion was to continue with two different MPI implementations, OpenMPI for Qlogic and MVAPICH for Mellanox.

OpenMPI was introduced in 2004 [9] and is an open source implementation of both the MPI-1 and MPI-2 specifications. The design is centered around the MPI Component Architecture (MCA) that is able to manage a wide variety of framework types such as Point-to-point Transport Layer (PTL), Point-to-point Management Layer (PML) or Collective Communication (COLL) [10].

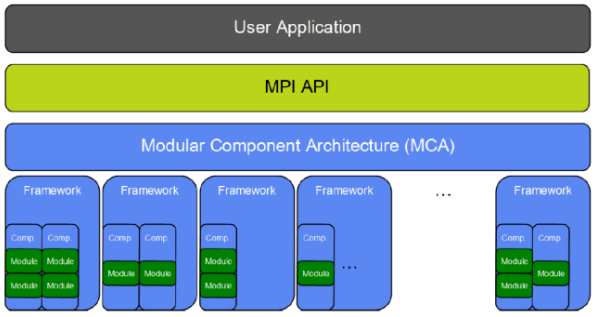

Figure 3: Modular Component Architecture (MCA)

Probably the most popular MPI implementation is **MVAPICH** which is based on the ADI interface of MPICH and was derived from MVICH. [11,12,13,14]. The first was available in 2001 at Ohio State University developed by Prof. Dhabaleswar Panda and it is now used by more than 800 organizations worldwide. Since 2009, the MPI-2 implementation based on the MPICH2 ADI3 layer is available including optimized support for two-sided and one-sided operations.

This study does not replace any benchmark study on a large system, it should be considered as a generic analysis of what performance numbers can be achieved in a small experimental setup with both types of technology. Especially adaptive routing mechanisms will not have any benefit in our setup.

3. Benchmarks

We used the micro-benchmarks for High-speed Interconnects designed by the Ohio State University [15] version 3.1.1.

The bandwidth measurements were carried out with the OFED tools:

```
OFED tools
==========
ib_write_bw -d mlx4_0 -a -n 5000
ib_read_bw -d mlx4_0 -a -n 5000
ib_write_lat -d mlx4_0 -a -n 5000
ib_read_lat -d mlx4_0 -a -n 5000
```

We noticed during the experiment, that a few environment mentioned in the MVAPICH user guide have a large impact on the performance. At the end we got the best results based on MVAPICH version 1.2.0 with these variables and numbers:

```
VIADEV_USE_COALESCE=1
VIADEV_COALESCE_THRESHOLD_SQ=1
VIADEV_PROGRESS_THRESHOLD=2
VIADEV_MAX_INLINE_SIZE=400
```

The following command was issued:

```
./osu_mbw_mr -w 512
```

For the Qlogic setup based on OpenMPI 1.4.3 no environmental variables or any specific options were used. We used IFS 6.0.

4. Results

We conducted the latency benchmarks in a ping-pong fashion. In figures 4 and 5 the results for the end-to-end latency in µs are shown for both test setups, Qlogic and Mellanox.

In both cases the smallest latencies were at 1.6 µs (Mellanox) and 1.7 µs (Qlogic) at small messages. For messages larger than 256B the latency numbers for Qlogic grow faster compared to Mellanox. In the range of messages from 256B to 4K, Qlogic shows up to 25% better results.

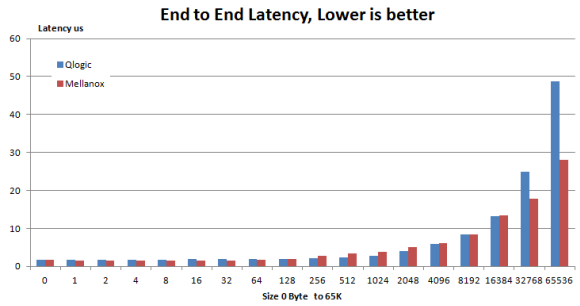

Figure 4: Latency for small messages

The picture changes again for larger messages as shown in figure 5. The average latency for Qlogic is 27% higher compared with the Mellanox numbers for messages between a size of 128K and 4MB.

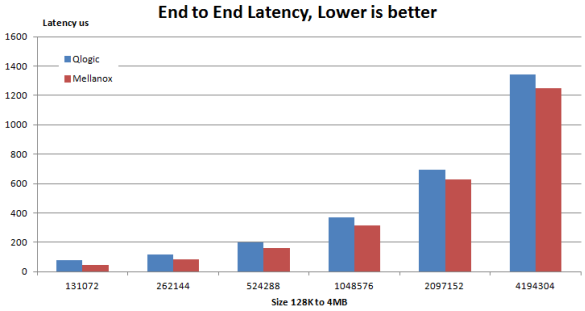

Figure 5: Latency for large messages

Figures 6 and 7 summarize the bandwidth measurements between two nodes based on the OSU benchmark for Mellanox and Qlogic.

For small messages between 128Byte and 4K, the results for Mellanox are significantly better than for Qlogic. At 512Byte the measured bandwidth exceeds the Qlogic number by a factor of 1.8.

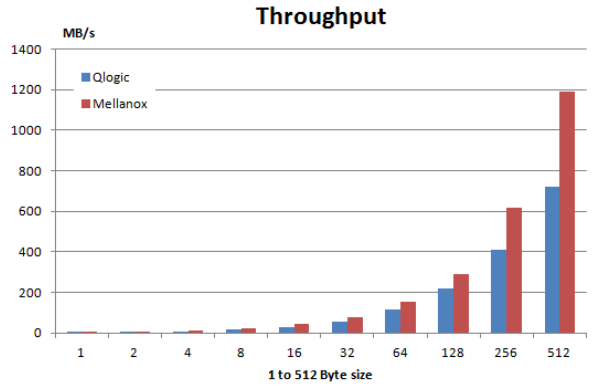

Figure 6: Bandwidth comparison for small messages

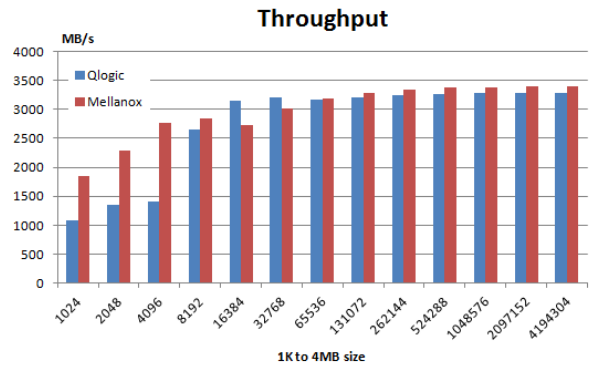

Figure 7: Bandwidth comparison for large messages

For message sizes larger than 4K the bandwidth measurements are almost equal with some small advantage for Qlogic at 16K and slightly better numbers for Mellanox by up to 4% for larger messages.

The figures also demonstrate that for messages larger than 16K, the MPI communication saturates the peak bandwidth values.

Figures 8 and 9 give the results for the message rate measurements. For messages

smaller than 128K the Mellanox setup yields up to 68Mio/s which is more than published anywhere else. To some extent, the environment variables mentioned before contribute to this outstanding result. Also this setup clearly outperforms the Qlogic setup with numbers at around 20Mio/s at most.

For message sizes above that size the differences in the measured numbers are negligible, with some advantages between 2K and 32KB for Qlogic with up to 15% higher message rates.

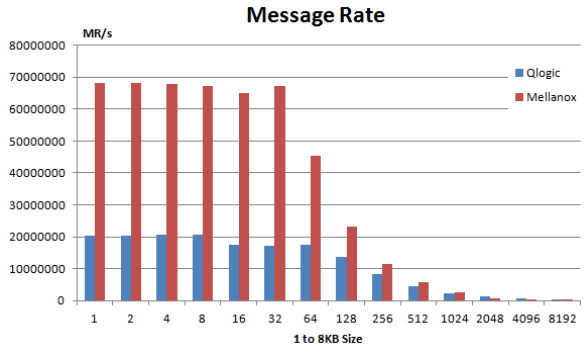

Figure 8: Message rates comparison for small messages

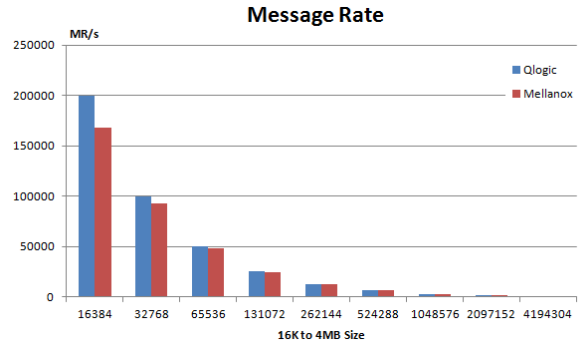

Figure 9: Message rates comparison for large messages

We also measured the dependency of the results on the processor frequency for small messages. For 1Byte to 16Byte the measured message rates are directly proportional to the processor frequency and reached nearly 80Mio Messages per second.

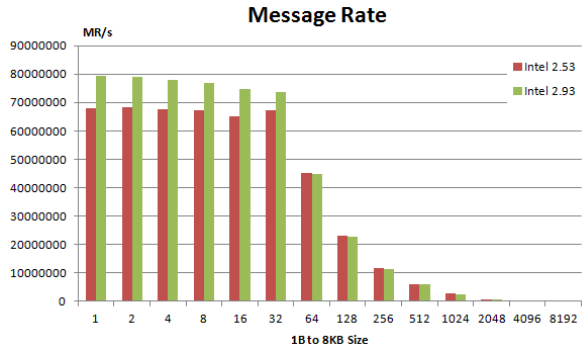

Figure 10: Message rate measured with Mellanox and different Intel processors.

Since Mellanox recommends MVAPICH as their preferred MPI implementation, we carried out all tests based on MVAPICH. However we were interested in the impact of a different MPI-implementation in the results. We repeated the message rate benchmark based on OpenMPI for the Mellanox setup. The results are given in figures 11 and 12.

The maximum message rate is limited to 10Mio messages/sec for OpenMPI for block sizes up to 128Byte and is far below the numbers measured with MVAPICH. For blocks larger than 128Byte, the picture changes completely and the message rates gathered with OpenMPI are much higher, for some of the block sizes an improvement by a factor of 11 was measured.

Both implementations use the *eager* algorithm for small messages and send the data and messaging metadata to an anonymous buffer on the target process, which later performs message matching and copies the data to the user buffer. Obviously this eager communication mechanism design varies between MVAPICH and OpenMPI and causes the differences in the observed message rate performance.

Taking the increasing trend of using one-sided and asynchronous communication

primitives such as put or get or other programming models like PGAs or UPC into account, real applications built on asynchronous communication models might take advantage of these extremely high message rate results.

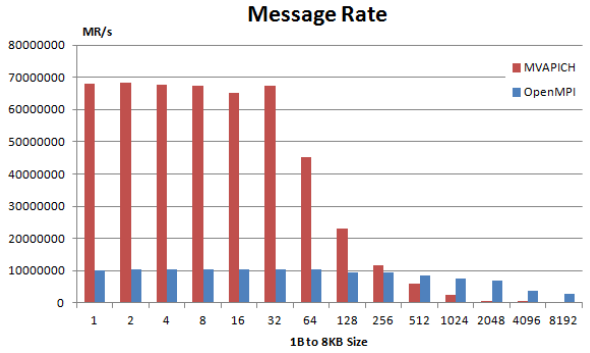

Figure 11: Impact of different MPI implementations on benchmark results (example for Mellanox setup) – small block sizes.

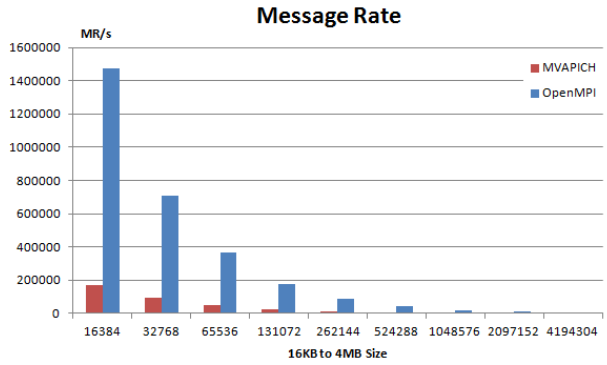

Figure 12: Impact of different MPI implementations on benchmark results (example for Mellanox setup) – large block sizes.

5. Conclusion

In this paper we have presented a performance comparison in a two-node experimental setup using Qlogic and Mellanox end-to-end solution using micro-benchmarks.

The study gives a few insights of the performance of both technologies and the impact of the MPI-implementation on the performance. The Mellanox setup in combination with MVAPICH outpaces Qlogic by far in the aspect of handling high message rates. Even nearly 80Mio MR/s have been measured.

Apart from that, the differences between both vendor products are not huge. On average, the latency and bandwidth results for the Mellanox setup are slightly better than those for Qlogic, though there are some cases, where Qlogic passes Mellanox.

For the real application performance the MPI-implementation is a very important part of the end-to-end solution.

Finally the application communication pattern and the efficiency of the optimization tools to improve the network communication within the fabric will be the decisive factor.

6. Literature

[1] Mellanox press, January 2011

[2] Mellanox Technologies, OpenFabrics Enterprise Distribution, Product Brief http://www.mellanox.com/related-docs/prod_software/PB_OFED.pdf 2010.

[3] Qlogic, InfiniBand Fabric Suite, Technology Brief, Technology Brief, 2011

[4] Demonstrate Performance, Scalability for HPC Applications , Press Release, June 2011.

[5] Qlogic, TrueScale InfiniBand, The Real Value, Technology Brief, 2009

[6] Addison Snell, Enabling Efficient Performance at Scale: QLogic IFS 6.0, Intersect360 research. White Paper, May 2010.

[7] ConnecX-3VPI, Mellanox Product Brief, website, 2011.

[8] IS5035, Product Brief, Mellanox, 2011.

[9] E. Gabriel et al., Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation, Proceedings , *11th European PVM/MPI Users' Group Meeting*, Budapest, Hungary, September 2004.

[10] Voltaire® User Manual for Open MPI v. 1.2, Release A00, July 2007.

[11] W. Gropp, E. Lusk, N. Doss, and A. Skjellum. A high-performance, portable implementation of the MPI message passing interface standard. Parallel Computing, 22(6), 1996. (MPICH)

[12] Ohio State University, MVAPICH2 User Guide, September, 2011

[13] Ohio State University, MVAPICH User Guide, January 2010,

[14] Lawrence Berkeley National Laboratory. MVICH: MPI for Virtual Interface Architecture. http://crd.lbl.gov/FTG/MVICH/mvich.shtml , August 2001.

[15] Ohio State University, Microbenchmarks, 2011