USENIX Association

Proceedings of the FREENIX Track:
2004 USENIX Annual Technical Conference

Boston, MA, USA
June 27–July 2, 2004

InfiniBand Performance Review: It's the Software Stupid

Troy R. Benjegerdes, Brett M. Bode
Scalable Computing Laboratory
Ames Laboratory
troy@scl.ameslab.gov, brett@scl.ameslab.gov
http://www.scl.ameslab.gov/

Abstract

The InfiniBand interconnect has received a great deal of attention recently as a result of its use in the Virginia Tech cluster that placed 3rd on the Top500 list. However, InfiniBand hardware and software have been available for over a year and continue to evolve. In addition, InfiniBand is supported by a number of different vendors each of which is seeking to differentiate themselves in the marketplace with slightly different hardware and software offerings. This paper will examine many of the current hardware and software implementations, illustrating their similarities and differences. In addition we will demonstrate the sometimes dramatic effects of the various tuning parameters of the various software implementations.

1 Introduction

The InfiniBand interconnect has seen a great deal of publicity in the past few years since its inception. Once the initial marketing faded many thought the whole concept had failed. In fact a great deal of effort was continuing to make the concept a reality. For a little over a year, 4X InfiniBand (10Gbps) hardware has been available from a variety of vendors. During that time the software stacks have matured a great deal to the point where it is now practical to use InfiniBand as the primary interconnect in a production oriented High-Performance Computing (HPC) system. Indeed the number 3 system on the top500 [ref top500.org] list is now an 1100 node cluster connected by InfiniBand.

This is not to say there is nothing left to be done. On the contrary, one of the biggest problems for the InfiniBand community is the software available. Up until early spring in 2004, each InfiniBand vendor was providing their own proprietary software stack. This software required a specific kernel binary from a distribution like Red Hat or Suse, and were generally only available x86 platforms. In our case, we have x86, amd64, and ppc64 platforms, so this was far from optimal. Recently, several vendors have released their hardware drivers and software stacks as open source. While this is a good step, it is much like the first releases of the Netscape source code as open source. Yes, it's out there and available, but it's not something anyone other than a dedicated hacker is really going to use.

Part of this review will cover the impressive performance results obtained in November of 2003, using vendor provided software stacks and MPI implementations. We will also examine performance of the latest low level InfiniBand driver stack from Mellanox, which is expected to be released soon under a dual GPL/BSD license. In addition, we will briefly examine application behavior of the GAMESS computational chemistry application on a 4 node InfiniBand cluster.

Finally, we will discuss some of problems with the currently available open-source InfiniBand stacks. Some of these problems include difficulty in the build process on systems not explicitly supported by the vendor, issues with multiple architecture support, and the disconnect between the larger network research community and the InfiniBand community.

2 History

First, some history of our experience with InfiniBand. Our first real exposure was at the InfiniBand Birds of a Feather at Ottawa Linux Symposium 2001 [InfiniBand BOF]. After this, we obtained InfiniBand HCA's and a switch development platform from Mellanox [Mellanox Technolgies].

At one point, both IBM and Intel had plans for making Host Channel adapters. Unfortunately, due to the 'dot-bomb' phenomenon, and other reasons known only to IBM and Intel decision makers, both companies canceled their plans for InfiniBand host adapters. However, Intel did keep on a team of software engineers on staff and open sourced their access layer. This code base was placed into a sourceforge project, and is generally referred to as the Intel Verbs layer, or IBAL [IBAL]. Unfortunately, since there was no shipping hardware, this access layer project got little attention outside of Intel.

2.1 InfiniBand finally delivers

Initially, Mellanox's low level drivers and firmware were still in early development releases, and performance was not that impressive. However, right before SuperComputing 2002, a succession of new firmware and drivers pushed the peak bandwidth achievable up to over 6 gigabits. This peak bandwidth was far above performance achievable on Myrinet, SCI/Dolphin, or 10 Gigabit Ethernet at the time. At this time, Dr. D.K. Panda's group at Ohio State University released their implementation of MPICH [MVAPICH] for Mellanox's VAPI InfiniBand software stack.

By SuperComputing 2003, several vendors, including InfiniCon had integrated together a software stack, based on the Mellanox low level drivers, an InfiniBand access layer, and MPI implementations based on MVIAPICH [InfiniCon]. There was also a large demand, particularly among the DOE laboratories and third party vendors, like Oracle, for Open Source InfiniBand drivers. At this point, all the major vendors were shipping products based on the Mellanox InfiniBand Host Channel Adapter (HCA). This led to vendors being reluctant to support an open-source solution, since they were attempting to differentiate and add value in the software stack.

2.2 Where's the source

The reluctance by vendors to open source was compounded by the fact that they were all using the same low level drivers provided by Mellanox, so even if they were to open source their access layer, in practice it would be useless without a low level driver. The IBAL sourceforge project currently still does not have a working open-source low level driver.

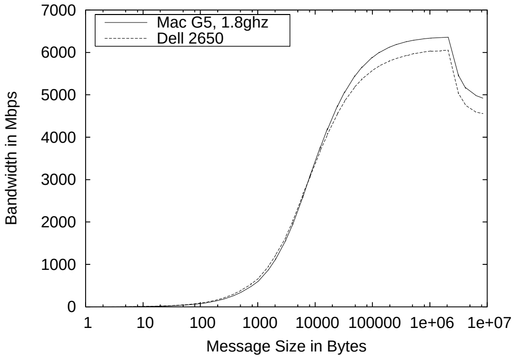

Figure 1: Raw Mellanox VAPI performance

Finally, in April of 2004, several InfiniBand vendors, including Topspin, InfiniCon, Voltaire, Mellanox, and Divergenet all announced, and released portions of their respective software stacks and drivers under various open-source licenses.The openib.org website was set up by a collaboration between InfiniBand vendors, and various US DOE Labs.

In some ways, the current situation resembles that of the initial Netscape source code release. There is a lot of code available for download from various places now, but none of it is something that can be used in a production cluster or data center environment. It is, however, a good opportunity for research.

3 Performance

Raw performance delivered by current Mellanox-based InfiniBand hardware is quite impressive and peak bandwidth is primarily limited by the PCI-X bus implementation on the system. Figure 1 illustrates the raw performance available using the Mellanox VAPI interface on a 2.4ghz Dell 2650 and 1.8ghz Macintosh G5.

3.1 Hardware test environment

For most of the data presented, the hardware test environment consisted of a cluster of 4 Dell 2650 dual Xeon systems, two Macintosh G5's, and two dual AMD Opteron systems. Mellanox-based HCA's from three vendors have been tested, and no noticeable performance difference has been noticed between vendors. All the software stacks tested have also worked on any HCA card with a Mellanox ASIC.

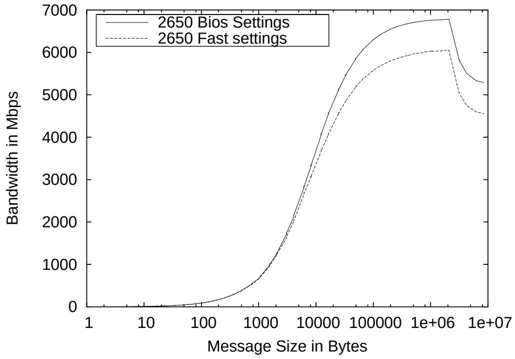

Figure 2: Serverworks chipset flag impact

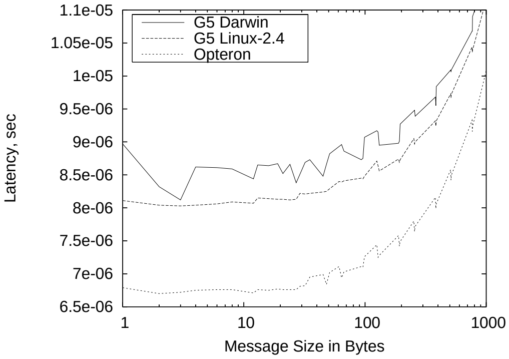

Figure 3: Raw VAPI latency with NetPIPE

The Dell 2650 systems consist of 2 2.2Ghz dual Xeon systems, and two 2.4Ghz dual Xeon systems purchased a year later, which have the faster 533mhz front-side bus. Most tests have been run on the 2.4Ghz machines. These systems have the ServerWorks Grand Champion chipset. These systems have a chipset flag which can be set by the Linux 'setpci' program which significantly increases peak PCI-X bandwidth, however it is reported setting this flag results in stability problems. Figure 2 shows the performance impact of setting, for lack of a better term, the Serverworks benchmark bit.

The Macintosh G5 test system consisted of a 1.8Ghz Mac G5, and a dual 2.0Ghz Mac G5. Performance is very similar to that of the AMD Opteron systems, which consisted of two Dual 1.4Ghz AMD Opterons with 8GB of memory each. This is expected since both systems use the AMD 8131 HyperTransport to PCI-X bridge chip. The only appreciable difference is the latency, which is to be expected since the G5 system has a bridge chip (the Uni-N ASIC) between the HyperTransport (HT) and CPU's, while the Opteron system has native HT links on the CPU. Figure 3 shows the difference in latency between the Opteron and G5 systems, taken from the November 2003 data. It is unclear why OSX(Darwin) has a higher latency than Linux on the same hardware.

3.2 Software test environment

Our first priority for our most recent round of InfiniBand performance tests was to reduce the overhead of installation and system maintenance. Due to this, we used the debian linux based NFS-root file system images we have already set up for other clusters. This requires that any InfiniBand drivers be buildable from source code, and run on the latest linux kernel (at this point linux-2.6.5, or linux 2.4.26), and work on the Debian linux distribution running from an NFS-root mounted filesystem. This quickly eliminated a number of the vendor provided solutions. The linux-2.4.26 kernel used for testing on the dell 2650's was also patched with Quadrics QSNet kernel patches due to testing Quadrics on the same machines.

Due to time constraints, we wound up only being able to run the pre-release thca-3.2-rc9, since this was the source base we had the most experience with trying to tweak it to run on our three types of test systems. In addition, with the exception of the Divergenet stack, all the other vendors are based on some Mellanox thca release, so this was the natural place to start. Due to the recent level of activity on openib.org, and internal vendor projects we expect we may have new results on other stacks by the time this paper is presented.

The AMD Opteron systems were running a debian-amd64 biarch system, with a 32 bit base system, and specific 64 bit libraries. All the InfiniBand libraries were built as 64 bit libraries since the Mellanox thca release does not have any facilities for biarch 32/64 bit environments, and the kernel code is 64 bit. Due to a build problem, we were unable to build a 2.6.5 kernel for Opteron at this time.

We were only able to obtain results on the Macintosh G5 on a 32 bit linux-2.4 kernel and on MacOSX, which is also 32 bit. After some changes, it is possible to get the Mellanox thca to build for a PPC64 linux environment, however the module does not load due to attempting to access a very low level memory management primitive. It is unclear if this is a generic InfiniBand issue, or something specific to PPC64. Even if

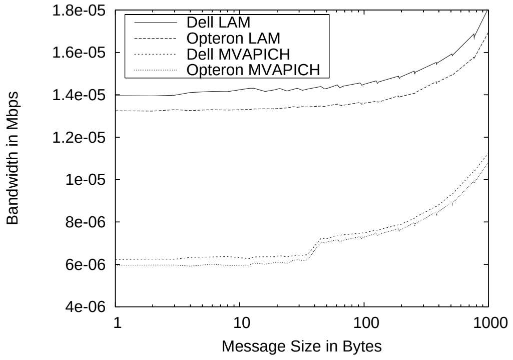

Figure 4: MPI Latency

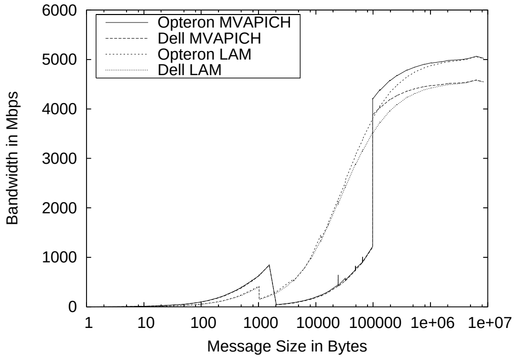

Figure 5: MVAPICH bad interaction with caches

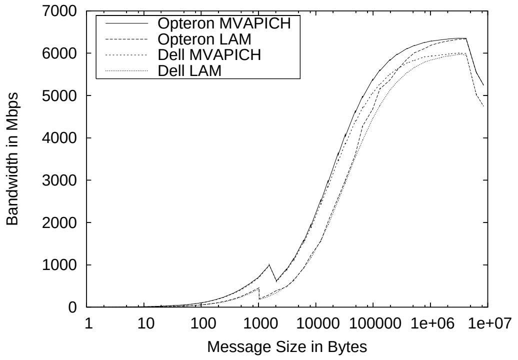

Figure 6: MVAPICH and LAM-MPI performance

the module were to load, it, it's not clear it would work due to issues with the PCI-X bridge and having to use an IOMMU. These problems are a rather strong indication that the current InfiniBand software was written primarily with x86 Intel systems and time-to-market considerations in mind rather than cross-platform software portability.

3.3 MPI testing

For MPI testing, we tested both the OSU-0.9.2 MVAPICH patches to MPICH-1.2.5, and an April 28 checkout of the LAM-MPI subversion repository. Figure 4 shows that MVAPICH, which uses an RDMA-write and memory polling, has significantly lower latency than LAM. MVAPICH also has a lower latency that NetPIPE-3.6's raw IB module, due to NetPIPE using POLL_CQ, which polls the InfiniBand card, causing extra PCI-X cycles.

The MVAPICH patches to MPICH are more mature, and have been available for over a year. They have been derived from the earlier M-VIA work at Berkeley Lab [M-VIA], and have several optimizations that LAM-MPI lacks. However, as Figure 5 shows, there are cases where maturity and advanced optimizations lose out to a simpler implementation.

These dropouts in MVAPICH only occur when running NetPIPE-3.6 with the '-I' cache-invalidate option, which causes NetPIPE to rotate through many buffers when sending ping-pong messages instead of re-using the same buffer. This forces worst-case behavior by causing a cache miss on every subsystem in the message path. In this case, there are internal caches in the MPI implementation for re-using so-called "eager" buffers, as well as a translation protection table (TPT) cache in the Mellanox ASIC. The TPT cache configuration is also the reason for the dropouts from around 6 gigabits to 5 gigabits on messages larger than 1-2MB on other graphs as well. Figure 6 shows the results of a NetPIPE run in which the messages are sent from the same buffers every time. MVAPICH makes effective use of internal caches and the TPT cache on the HCA as well, showing noticeable better performance than LAM-MPI at medium message sizes.

The dropouts in MVAPICH are not, however, inherent to the code base. Figure 7 shows the performance of the MVAPICH and InfiniCon MPI from our earlier November 2003 data. The code base is largely the same, since InfiniCon used MVAPICH as a base. The differences appear to result from differences in tuning parameters, interaction with the memory management subsystem, and the Mellanox TPT cache. With the cache-invalidate option, InfiniCon's MPI only has two small dropouts around 10K byte messages sizes, opposed to MVAPICH, which has a drop from around 2K to 100k. Differences in

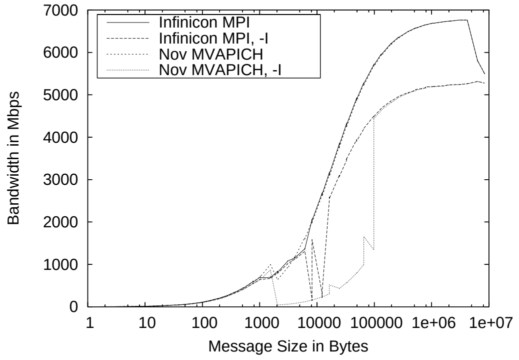

Figure 7: MVAPICH and InfiniCon MPI

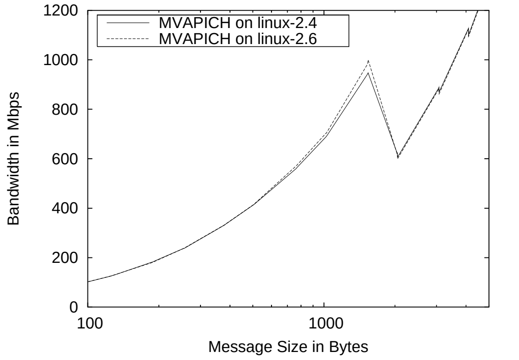

Figure 8: Kernel 2.4 vs 2.6, small messages

peak bandwidth from other graphs are due to the Serverworks chipset options discussed earlier.

3.4 Kernel versions

Figure 8 and Figure 9 show very little difference between a linux-2.4 series kernel, and linux-2.6.5 running on the Dell 2650 hardware. This is to be expected since the current driver implementations bypass many of the linux kernel subsystems.

4 Conclusions

We have seen that InfiniBand can achieve very good performance. What remains is to see how well it continues to evolve and compete with other custom cluster interconnects. Vendor support for small clusters and 'commercial' linux distributions like Red Hat and Suse seems to be very good. What's missing is better support for open source, and a common API for application development.

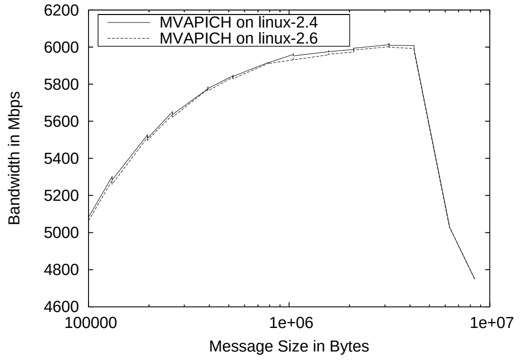

Figure 9: Kernel 2.4 vs 2.6, big messages

4.1 A Commodity Interconnect?

Many claims have been made over the history of InfiniBand about it becoming the 'new' commodity network interconnect. The first several years of it's existence were met with a great deal of skepticism from various communities that InfiniBand was just another marketing buzzword, and a bunch of vaporware. Now, the hardware exists, and proprietary driver stacks seem to work well in specific environments. Current pricing on HCA's is around $750 a port, and switch pricing is $300 a port. While this pricing is quite competitive for other cluster-specific interconnects, it is dependant on InfiniBand adoption in the Data Center and Enterprise Storage markets. It remains to be seen whether the potential of commodity volume production and raw performance is offset by the additional complexity of the software stack. 10 Gigabit Ethernet has a definite advantage in that drivers for the Intel 10Gig-E card have been in the linux-2.6 kernel series for several months, and other vendors have submitted drivers for inclusion into the mainline kernel.

4.2 Linux integration

InfiniBand's biggest (and some would say fatal) flaw is the amount of new code required to just get something to run. The Mellanox low level driver alone is over 100,000 lines of code. This doesn't count extras like sockets direct, SCSI Remote Protocol, or an IP over IB driver. We have some very nice OS-bypass hardware, but it requires what amounts to half an OS worth of additional software to run. The hoped-for commodity markets of scale will never occur unless adding InfiniBand drivers is not much different than adding an ethernet driver. In the case of Linux, this means integration into the memory management subsystem in a clean, cross-platform manner, and a minimal driver that can bring the card up and send packets without a lot management code.

There is a definite opportunity here to develop a clean API for RDMA-capable networks like InfiniBand, and get that API integrated into Linux. But it's got to be something for more than just InfiniBand. 10 Gigabit Ethernet is going to need some sort of RDMA capability to function well, and the existing high-performance cluster networks could benefit from something like this as well. The real benefit isn't necessarily to the network vendor, it's to application developers who currently use the Berkeley Sockets API, because it's the only thing that's portable. Sockets direct is appealing, but in order for it to work, there needs to be a consensus on how it is to work across different types of networks.

5 Acknowledgments

This work was performed under auspices of the U. S. Department of Energy under contract W-7405-Eng-82 at Ames Laboratory operated by the Iowa State University of Science and Technology. Funding was provided by the Mathematical, Information and Computational Science division of the Office of Advanced Scientific Computing Research.

We would also like to thank InfiniCon Systems for a hardware loan, Jeff Kirk at Mellanox for invaluable technical assistance, everyone contributing to the OpenIB.org project, and linux kernel developers who have taken an interest in InfiniBand.

References

[InfiniBand BOF] InfiniBand on Linux BOF, *http://www.linuxsymposium.org/2001/bofs.php* Ottawa Linux Symposium, 2001

[Mellanox Technolgies] *http://www.mellanox.com*

[IBAL] *http://infiniband.sourceforge.net*

[MVAPICH] *http://nowlab.cis.ohio-state.edu/projects/mpi-iba*

[InfiniCon] *http://www.infinicon.com*

[M-VIA] *http://www.nersc.gov/research/FTG/via/download_info.html*