

First Experiences with the InfiniBand Interconnect

Ulrich Schwickerath^a, Andreas Heiss^b

^a Institute for Scientific Computing (IWR), Forschungszentrum Karlsruhe GmbH
P.O. Box 36 40, 76021 Karlsruhe, Germany

^b Institute for Scientific Computing (IWR), Forschungszentrum Karlsruhe GmbH
P.O. Box 36 40, 76021 Karlsruhe, Germany

A test cluster of dual Intel-Xeon processor server nodes has been equipped with 10 GBit/s InfiniBand hardware. Capabilities of this new interconnect were tested and compared to Gigabit - Ethernet (GE) hardware with respect to both High Performance Computing (MPI based parallel computing applications) and High Throughput Computing (HTC). RFIO, a protocol for fast and efficient file transfers, has been ported to make immediate use of the InfiniBand hardware, utilizing the remote direct memory access (RDMA) capabilities of the InfiniBand hardware. The performance is compared to Gigabit-Ethernet.

1. Introduction

The steadily increasing performance of commodity CPU's during the last years has increased the interest in cheap, off-the-shelf PC clusters. Usually, the CPU's in such clusters communicate via cheap, nowadays often on-board, Ethernet interfaces. However, networking speed did not increase in the same way as the CPU power did. On the other hand, High Performance Computing (HPC) applications which do a lot of communication require low latency and high bandwidth. A large bandwidth is also required for High Throughput Computing (HTC) applications like cluster file systems[1]. InfiniBand[2] offers an open standard which matches these requirements. Hardware adapters at 10 GBit/s speed are available on the market since late summer 2002. For HPC applications several free and commercial MPI ports exist, see for example [3,4]. The maturity of the technology is well advanced nowadays. At the time of this writing the Virginia Tera Scale Cluster which uses 10 GBit/s - InfiniBand is ranked the third fastest supercomputer system in the world[5], and was build at very low cost.

The IWR has started its InfiniBand evaluation project in autumn 2002. A test cluster has been set up during the year 2003, and real world

MPI applications are being tested on it. Within the context of HTC, the RFIO protocol, which is part of CERN's Advanced Storage Manager (CASTOR) project[7] has been ported to the InfiniBand Verbs API as implemented by the hardware vendor Mellanox[8].

2. InfiniBand

The InfiniBand specifications[2] define an open standard for a scalable, channel oriented, low latency and high bandwidth interconnect at three different speeds of 2.5 GBit/s (1x), 10 GBit/s (4x) and 30 GBit/s (12x). InfiniBand provides a channel oriented, switched fabric for both inter-processor and I/O communications. The fabric can be set up in a redundant way, and Quality of Service (QoS) is supported. Both copper and fiber cables are foreseen. The hardware is remote DMA capable. For reliable connections the hardware takes care of the data integrity. The transport layer is queue based. A send and a receive queue make up a queue pair (QP) which can be used for data transfers. Completion events are posted into specific completion queues (CQ). The specifications also define a set of Verbs; however, the detailed implementation of the Verbs functions is vendor specific. In this study the Mellanox Verbs interface (VAPI)[8] has been used.

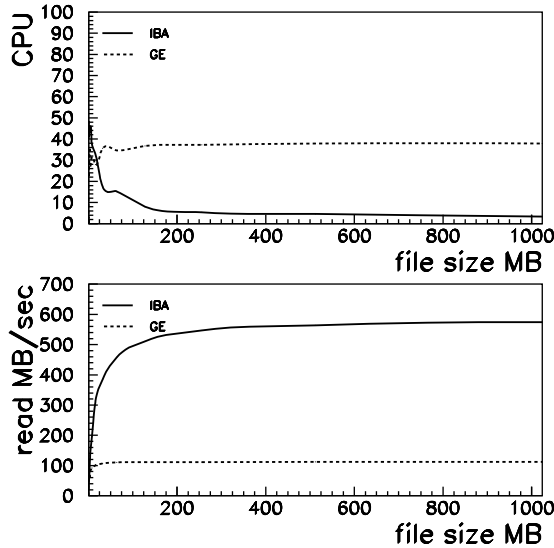


Figure 1. Reading garbage in streaming mode with rfc and dropping the output.

3. Hardware setup

The test equipment is divided into two parts: a small cluster made of eight worker nodes (dual Intel - Xeon boxes running at 2.4 GHz, with RedHat Linux 7.3) is used for evaluation of real life MPI applications. An additional server is used for interactive login sessions. All nodes are connected by a 4x (10 GBit/s) InfiniBand fabric with a 16 port fat tree 4x-InfiniBand switch. Another three nodes are used for fabric management, hardware evaluation and software development. These nodes are connected to a switch card providing another 6 external ports. A two port 2 GBit/s Fiber Channel (FC) to InfiniBand card is used to connect two IDE raid boxes to the system. For very large files, the read(write) performance of each raid array is about 250(110) MB/s, estimated with bonnie++[6]. For smaller files, the performance is larger due to caching effects.

The full system of 24 CPUs has a performance of about 70 GFlops, measured with the HPL

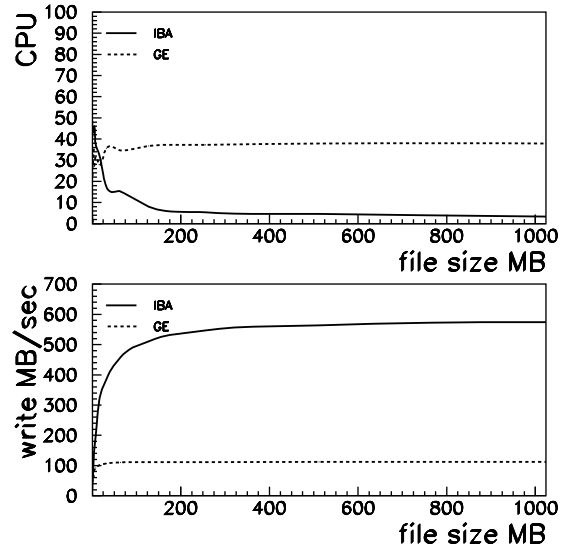


Figure 2. Writing garbage in streaming mode with rfc and dropping the output.

benchmark. The scalability of the system up to the 24 processors is good.

4. RFIO over InfiniBand

4.1. RFIO

The Remote File I/O (RFIO) protocol is under development since 1990 when it was part of the SHIFT project at CERN. Nowadays it is part of the CERN Advanced Storage Manager (CASTOR) project[7]. CASTOR provides transparent tape access through a Hierarchical Storage Manager (HSM) system, and relies on the RFIO protocol. RFIO itself consists of

- a daemon (rfiod)
- an application interface library (API) with POSIX like functions
- a couple of useful applications for remote file access

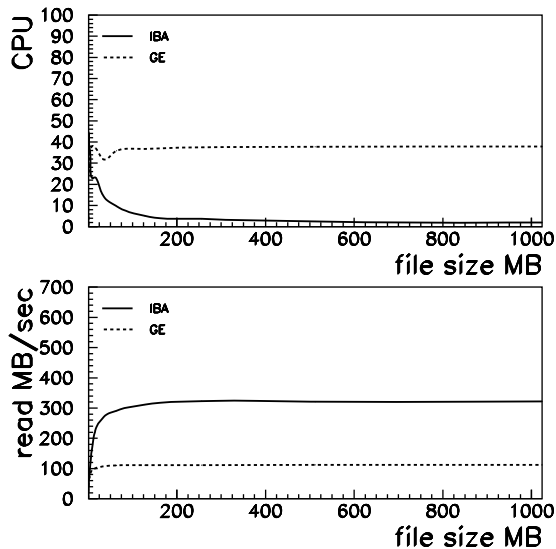


Figure 3. Reading a remote cached file in streaming mode with rfcv and dropping the output.

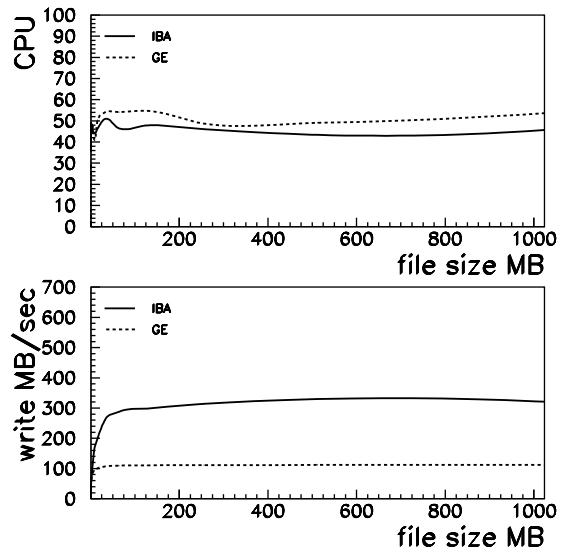


Figure 4. Writing a remote cached file in streaming mode with rfcv and dropping the output.

and provides fast and efficient access to files on remote hosts. Remote file access can be random access or in a streaming manner. For sequential access to files which occurs for example in simple file copies a special fast streaming protocol is implemented.

4.2. The InfiniBand patches

The InfiniBand patches for RFIO make use of the possibility to have reliable connections and RDMA access to data on a remote host. This combination allows to reduce the CPU consumption in transfers of large files. Addressing of the remote host is done via Ethernet for transparency. As soon as a server is contacted, the availability of InfiniBand is checked, connection informations are exchanged, and all further communications are done via the InfiniBand link. Else, a fall back to Ethernet is done. For sequential access to a remote file a new streaming protocol has been implemented. In this case two queue pairs are being used. The QP that has been established

at request time is used only for exchange of control messages. A second QP is used for bulk data transfers. Buffers are filled using RDMA write requests over a reliable connection between server and client.

5. Results

Since the time of the ACAT03 conference, where preliminary performance numbers were presented, a couple of problems have been fixed, and the measurements have been redone. The following results are based on updated patches for CASTOR version 1.6.1.3 from December 2003. For each measurement point the mean value of 100 sequential single measurements has been used. The time and the CPU consumption have been measured with the shell build-in time command which is sufficiently precise for large file sizes. Because of the need to exchange connection informations like memory locations and access keys when a new connection is established,

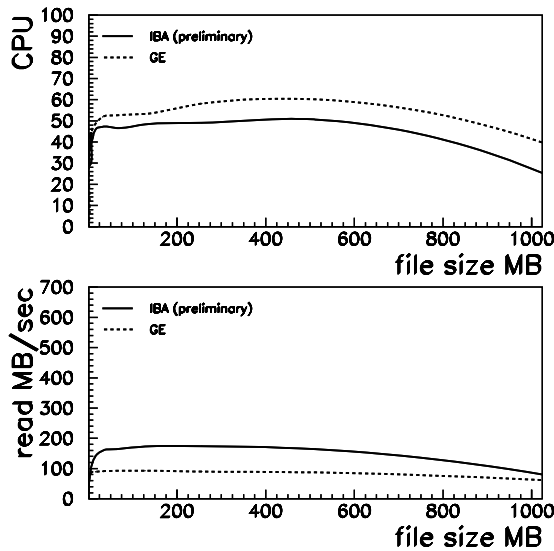


Figure 5. Reading a remote file and writing it to locally attached IDE disk arrays.

the performance is better for larger files than for small ones, in particular in terms of CPU usage. Furthermore, the precision of the time command is limited resulting in unreliable measurements for very small file sizes. Gigabit-Ethernet numbers were taken using a cross cable between the two hosts used for the measurements. All measurements were done on the same two nodes. The RDMA raw transfer speed on the test cluster is about 780 MByte/s. The RFIO protocol adds some overhead which reduces the actual transfer speed. The protocol plus network performance can be estimated if the transferred buffers are not filled with meaningful data. This way, bottle necks due to bus speed between processor and main memory can be avoided. In Fig. 1 and 2 the read and write performance is shown. The peak performance is reached for large file sizes, and reaches up to 550 MB/s at very low CPU consumption. For GE, the performance is limited by the network speed, and reaches about 110 MB/s. In Fig. 3 and 4, the data source are files which

are cached in memory. The files are truly transferred, but the output is dropped. Peak performance drops to about 300 MB/s, both for read and write. The CPU usage for remote writes is larger than for reads because in this case the client has to do a `read()` system call to fill the buffers to be written to the network. This involves memory copy operations which require some CPU cycles. The drop in performance as compared to the Fig. 1 and 2 points to a bottle neck in the server architecture. This interpretation is supported by a measurement done on an Intel Itanium 2 system at CERN which is equipped with 4x-InfiniBand, that showed a peak transfer rate of about 450 MB/s. For GE the transfer speed is limited by the network bandwidth. Fig. 5 shows an example of reading a remote, cached file, and writing it to a locally attached IDE disk array. Here, the transfer rate is limited by the disk array write performance. The disks are fast enough to saturate the GE link.

REFERENCES

1. J. Wu, P. Wyckoff, and D. K. Panda, PVFS over InfiniBand: Design and Performance Evaluation, ICPP03, Oct. 6-9, 2003.
2. InfiniBand Trade Association, InfiniBand Architecture Specification, Release 1.0, October 24, 2000.
3. J. Liu, J. Wu, S. P. Kini, P. Wyckoff, and D. K. Panda, High Performance RDMA-Based MPI Implementation over InfiniBand, Int'l Conference on Supercomputing (ICS '03), June 2003.
4. MPICH device for InfiniBand, diploma thesis, Rene Grabner and Frank Mietke, August 2003, Chemnitz University of Technology, Germany.
5. 22nd Edition of TOP500 List of World's Fastest Supercomputers, <http://www.top500.org>
6. Russell Coker, Bonnie++, <http://www.coker.com.au/bonnie++>
7. CASTOR (CERN Advanced Storage Manager) <http://castor.web.cern.ch/castor>
8. Mellanox Technologies. Mellanox IB-Verbs API (VAPI), Rev. 0.98, August 2003.