ISSN No. 0976-5697

Volume 6, No. 8, Nov-Dec 2015

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Performance Analysis of High Computational Jobs using Infiniband Interconnect

Jaswinder Singh
Computer Science & Engineering
Northwest Institute of Enggineering and technology
VPO. Dhudike, Moga(Pb), India.

Dr. Mohita Garg
Computer Science & Engineering
Northwest Institute of Enggineering and technology
VPO. Dhudike, Moga(Pb), India.

Abstract: The aim of this paper, analyzing the performance of Ethernet using matrix multiplication of higher order with the help of infiniband interconnects system. Second improving the capacity of Ethernet bandwidth from 10 Mbps to 40 Gbps. This paper described the Matrix Multiplication problem which is taken and contain both multiplication and addition factor operations for output of the results. Results will classify the job size to be taken on both interconnect for optimal use of the bandwidth. The whole proposed work will required in open source Linux using MPI. The developed interconnected model has been work in efficient way and it has been shown that this model is faster than the previous infiniband interconnected system and the bandwidth is induced.

Keywords: Bioinformatics, mpiBLAST, HPC, Infiniband and Cluster Computing.

I. INTRODAUCTION

A parallel system consists of two or more tightly coupled processors, typically of the same type, connected by some form of communication network. An effective parallel system requires an appropriate parallel machine and a well-optimized parallel program. When a long series of identical computations is to be performed, such as those required for the formation of numerical tables, the machine can be brought into play so as to give several results at the same time, which will greatly abridge the whole amount of the processes.

Similarities between newly discovered sequence and a known sequence can help in determining functions of the new sequence and find sibling species from common ancestor.

There are two types of sequence alignment problems:

- First global alignment algorithm finds the best match between the entire sequences.
- Second local alignment algorithm finds the best match between parts of the sequences.

The first algorithms devised for sequence-alignment were Needleman Wunsch (1979) and Smith Waterman (1981). These were based on dynamic programming and produce optimal solutions but had time complexity O (n2). [5]

Modern high-end Personal Computers (PCs) provide computation speed as well as storage capacity at the best price/performance ever. Therefore an obvious way to obtain higher performance, parallel systems has to build a distributed memory platform out of a pool of PCs interconnected by a fast Local Area Network (LAN) hardware, commonly called a cluster of PCs. Local Area Network provides huge amount of unused computational power that can be tapped to solve large complex problems in parallel. However standard communication protocols like TCP/IP run at a very low efficiency rate on modern LANs. This has the immediate consequence of poor performance achieved by parallel programs on clusters.

Its capacity improve from 10 Mbps to Gigabit, where as Infiniband provides bandwidth of 40 Gbps. Matrix Multiplication problem is taken as it contains both multiplication and addition factor for output of the results

II. RELATED WORK

In this research, we discover that Infiniband is new developed interconnect for cluster computing. Ethernet has grown from 10Mbps to Gigabit but still lack the data transfer in case of mission critical problems. Matrix Multiplication is taken as compute intensive example to check the performance of the middleware of cluster system. It contains both addition and multiplication operations.However standard communication protocols like TCP/IP run at a very low efficiency rate on modern LANs. This has the immediate consequence of poor performance achieved by parallel programs on clusters. Some related working ways are described bellow:-

A. Cluster paper

Hardware configuration for the cluster formation is the basic requirement for the computation of the parallel program for matrix multiplication.This research discovers that a long bandwidth is not able to send or share proper data in parallel system. There are a lot of algorithms and related working platforms are available to describe the parallel processing system. Some related working algorithms are described bellow:-

In this research, the concept of message passing on cluster computers with high performance is an essential paradigm for computation. High performance computing tasks are use cluster of personal computer which becomes very effective and popular, because they support a high performance to a much lower price compare to dedicated multicomputer.

According to a research work a typical example is, if PC A is waiting to receive information from PC B, while B is also waiting to receive information from A. The matrix operation derives a resultant matrix by multiplying two input matrices, a and b, where matrix a is a matrix of N rows by P columns and matrix b is of P rows by M columns. The resultant matrix c is of N rows by M columns. Its algorithm requires n3

multiplications and n3 additions, leading to a sequential time complexity of O (n3).

In a Master-Slave computing paradigm, a Master process takes the work performed in the computationally intensive loop and divides it up into a number of tasks that it deposits into a task bag.

One or more processes, known as slaves, grab these tasks, compute them and place the results back into a result bag. The Master process collects the results as they are computed and combines them into something meaningful such as a vector product [1].

B. HPC (High-Performance Computing)

High-performance computing (HPC) is the use of parallel processing for running advanced application programs efficiently, reliably and quickly. The term applies especially to systems that function above a teraflop or floating-point operations per second. The term HPC is occasionally used as a synonym for supercomputing, although technically a supercomputer is a system that performs at or near the currently highest operational rate for computers. Some supercomputers work at more than a pet flop or floating-point operations per second [8].

- There are a large number of HPC applications that need the lowest possible latency for best performance or the highest bandwidth.
- 10GigE has 5-6 times the latency of InfiniBand
- InfiniBand has 3.7x the throughput of 10GigE
- Beyond 1-8 nodes, many times InfiniBand pro¬vides much better performance than 10GigE and the performance difference grows rapidly as the number of nodes increases
- There are some HPC applications that are not latency sensitive. For example, gene sequencing and some bioinformatics applications are not sensitive to latency and scale well with TCP based networks including GigE and 10GigE. For these applications, both 10GigE and InfiniBand are appropriate solutions
- Putting HPC message passing traffic and storage traf¬fic on a single TCP network may not provide enough data throughput for either. Many HPC applications are IOPS driven and need a low-latency network for best performance. 10GigE networks have 3-4 times the latency of InfiniBand. For most HPC applications it is recommended to use InfiniBand when storage and computational traffic is combined.
- There are a number of examples that show 10GigE has limited scalability for HPC applications and InfiniBand proves to be a better performance, price/ performance, and power solution than 10GigE.[2]

C. INFINIBAND

InfiniBand is a powerful new architecture designed to support Input and output connectivity for the Internet infrastructure. InfiniBand is supported by all the major OEM server vendors as a means to expand beyond and create the next generation I/O interconnect standard in servers. For the first time, a high volume, industry standard I/O interconnect extends the role of traditional “in the box” busses. InfiniBand is unique in providing both, an “in the box” backplane solution, an external interconnect, and “Bandwidth Out of the box”, thus it provides connectivity in a way previous lyre served only for traditional networking interconnects. This unification of I/O and system area networking requires a new architecture that supports the needs of these two previously separate domains [3].

An InfiniBand fabric has the characteristics that make it well suited for use as a unified fabric. For example, its message orientation and its ability to efficiently handle the small message sizes that are often found in network applications, as well as the large message sizes that are common in storage. In addition, the InfiniBand Architecture contains features such as a virtual lane architecture specifically designed to support various types of traffic simultaneously. [9]

InfiniBand simplifies application cluster connections by unifying the network interconnect with a feature-rich managed architecture. InfiniBand’s switched architecture provides native cluster connectivity, thus supporting scalability and reliability inside and “out of the box”. Devices can be added and multiple paths can be utilized with the addition of switches to the fabric. High priority transactions between devices can be processed ahead of the lower priority items through QoS mechanisms built into InfiniBand.[3]

D. THE MPIBLAST ALGORITHM

MPIBLAST is a freely available open-source parallelization of National Centre for Biotechnology Information (NCBI) BLAST, which achieves super linear speedup by segmenting a BLAST database. It is designed to work on a computer cluster using MPI library and adopts a master-slave style. (Darling et al 2003) The mpiBLAST algorithm consists of three steps:

1. Segmenting and distributing the database,
2. Running mpiBLAST queries on each node,
3. Merging the results from each node into a single output.

Before mpiBLAST search, the database is formatted and segmented using a wrapper called mpiformatdb and placed at shared storage. mpiBLAST enables the master node to assign the query sequence and database fragment to each worker node. The worker nodes perform the BLAST search on queries and send the results to the master node. When one worker node complets its task, the master node assign a new fragment to it. This procedure is repeated until all the queries have been searched. The master node merge all the results and sorts them according to score. Results written in output file can be in any format including XML, HTML, simple text, ASN.1. However, mpiBLAST suffers from non-search overheads with increasing number of processors and varying database sizes.

BLAST that stands for parallel I/O BLAST and uses MPIIO for efficient data access. MPI-IO enables multiple processors to read or write files simultaneously. (Correa and Silva 2011) One of pioBLAST's main updates was the caching of sequences by worker nodes as they find potential alignments in their partial results.

Due to parallel writing of output, pioBLAST greatly improved the performance. As a result, some of the pioBLAST's enhancements were added to mpiBLAST, resulting in the development of mpiBLAST-PIO, which is the official version of mpiBLAST since release 1.6 (Lin H et al 2005). mpiBLAST-PIO is optimized and extended version of parallel and distributed-memory version BLAST. The extensions include a virtual file-manager, a “multiple master” runtime model, efficient fragment distribution and intelligent load balancing. [5]

III. MPI IMPLEMENTATIONS

The proposed mpiBLAST, an open source parallelization of BLAST database and then having each node in a computational cluster search a unique portion of the database. Database segmentation permits each node to search a smaller portion of the database, eliminates disk I/O and vastly improves the BLAST performance.

The Message Passing Interface (MPI) is the dominant programming model for parallel scientific applications. Given the role of the MPI library as the communication substrate for application communication, the library must ensure to provide scalability both in performance and in resource usage. In our experiments, we used two of the most commonly used MPI implementations in the HPC industry, which are MVAPICH2 and Intel MPI. - Intel MPI Based on MPI-2 specifications, Intel MPI Library focuses on making applications perform better on Intel architecture based (AI) clusters. This MPI implementation has the ability to function on multiple HPC fabrics using an accelerated universal, multi-fabric layer for fast interconnects via the Direct Access Programming Library (DAPL) methodology [15]. Thus, developers can deal with MPI codes, independent of the fabric, knowing that it will run on whatever fabric is chosen at runtime. In addition, Intel MPI supports various runtime environments and modules to integrate with other HPC job schedulers, such as Load Sharing Facility (LSF) by Platform Computing and Torque scheduler that is provided by Cluster Resources. [7]

All MPI programs must call MPI_INIT as the first MPI call, to initialize themselves.

Most MPI programs call MPI_COMM_SIZE to determine the size of the current virtual machine, that is, how many processes are running.

Most MPI programs call MPI_COMM_RANK to determine their rank, which is a number between 0 and size-1.

Conditional process and general message passing can take place, for example, using the calls MPI_SEND and MPI_RECV.All MPI programs must call MPI_FINALIZE as the last call to an MPI library routine. [6]

IV. CONCLUSIONS

In the present study, performance analysis using matrix multiplication of high order is done for Ethernet and Infiniband interconnect. Ethernet has improved it capacity from 10 Mbps to Gigabit, where as Infiniband provides bandwidth of 40 Gbps. Matrix Multiplication problem is taken as it contains both multiplication and addition factor for output of the results. Results will classify the job size to be taken on both interconnect for optimal use of the bandwidth.

V. ACKNOWLEDGMENT

This study was conduct by the first author under the direction of the co-author in fractional accomplishment of the necessities of a Master Degree in Computer science and Engineering. The First author wishes to thank Dr. Mohita Garg HOD of Computer Science and Engineering at NWIET Institute of Engineering and Technology, Dhudike, Ajitwal Moga, under Punjab Technical University , Jalandhar for his support over the period in which this article was written.

VI. REFERENCES

[1] Sherihan Abu ElEnin, Mohamed Abu ElSoud ,"Evaluation of Matrix multiplication on an MPI Cluster", IJECS-2011.

[2] Interconnect Analysis,"10GigE and InfiniBand in High Performance Computing",2009.

[3] " Introduction to InfiniBand",2009.

[4] Zach Hill* and Marty Humphrey ,"A Quantitative Analysis of High Performance Computing with Amazon's EC2 Infrastructure: The Death of the Local Cluster?",2009.

[5] Nisha Dhankher , O P Gupta ,"Parallel Implementation & Performance Evaluation of Blast Algorithm on Linux Cluster,2014.

[6] Javed Ali ,Rafiqul Zaman Khan, "Performance Analysis of Matrix Multiplication Algorithms Using MPI,"2012.

[7] Javed Ali ,Rafiqul Zaman Khan," Performance Benchmark and MPI Evaluation Using Westmere-based Infiniband HPC Cluster,"2003.

[8] Ye Xiaotao1, Lv Aili1, Zhao Lin2," Research of High Performance Computing With Clouds," 2010.