On the Timing Analysis of Cluster based Communication Devices for Large Scale Computing Systems

Mohammed Mahfooz Sheikh, A M Khan*
Dept. of Electronics,
Mangalore University,
Mangalagangotri – 574199,
Karnataka, India.
*amkhan@mangaloreuniversity.ac.in

U N Sinha
Distinguished Scientist
CSIR C-MMACS/NAL
Bangalore, India - 560037

Abstract— **Many parallel computing environments utilize cluster based architecture for large scale computing owing to the ease of their availability. As the cluster based approach may be used extensively, the interconnection mechanism plays a vital role in the performance of the system. The globally coupled class of problem is generally not amenable with the cluster based approach due to its substantial demand for communication across the architecture. In this paper we present a timing analysis of standard cluster based communication devices viz, Ethernet, InfiniBand and the custom designed Floswitch.**

Keywords- Ethernet, InfiniBand, Floswitch, Cluster Computing.

I. INTRODUCTION

Parallel Processing has become an inevitable tool for solving complex scientific problems that involve large scale computations. Without large scale computational resources genome sequencing could not have been possible [1]. Without the help from primitive computer, design of atom bomb would not have been feasible [2]. New drug development routinely uses large scale computing [3]. Many new discoveries have been result of large scale computations. For example, solitary waves were found by Ulam and his colleagues using large scale computing [4]; space missions demand massive computing for re-entry trajectories of space vehicles and numerical precision exceeding 20 digits are quite common. It is, therefore, not surprising that requirement of large scale computations has led to development of parallel machines with history dating back to 1960s [5], [6]. The story of developments of the computers in use till early 70s is well documented and vividly presented in the references [7]–[12]. Parallel machines are generally built by the interconnection of more number of processors and their architectures purely depend upon the complexity of the tasks which demands the type of coupling required. The parallel processing tasks are divided among various Processing Elements (PEs) that execute the jobs in parallel. It is implicitly assumed here that the task is agreeable with parallel processing architecture and the communication mechanism is in place so that PEs may work on the subtasks of the main task. Yet they would complete the main task as if the process is carried out on a single virtual sequential computing machine. Communication paradigm appears at a cross road at this point. It is a fact, that the field equations occurring in science when appropriately formulated very well requires distributed parallel processing. A simple example will illustrate the view point. The solution of the potential equation which is formulated through Greens function is not naturally amenable to parallel processing whereas when formulated by finite difference discretisation leads naturally to domain decomposition technique which is highly amenable to parallel processing [13]. The PEs in the parallel machines are thus required to cooperate to solve a particular task needing interconnection scheme for communicating with each other. Such environment offers faster solution to complex problems than feasible using sequential machines. Moreover sequential machines may not be able to solve the problem in reasonable amount of time. The interconnection network required for the PEs to communicate forms the most important part of a parallel computer next to Central Processing Units (CPUs).

II. COMMUNICATION TECHNOLOGY IN PARALLEL COMPUTING

It is seen that as the power of microprocessor based CPU keeps growing, the tendency to put them together for building bigger and bigger parallel computers does not decline; this gave rise to the development of a host of interconnection networks ranging from shared memory devices to crossbar switch, Ethernet, and InfiniBand type connectivity. The supporting software also grew in functionalities and ease of operation. MPI [14] (Message Passing Interface) is one of them, and is commonly used in academic and research institutes.

Shared memory architecture and crossbar switch technology are technically sophisticated and also quite expensive. Therefore, it is not surprising to find that cluster computing based on the model of connecting a large number of PCs and workstations through Ethernet [15], InfiniBand [16] or proprietary connections [17] emerged as early as mid-nineties. The term Beowulf was very fashionable at that time and the book of Sterling *et al*. [18] from MIT devotes significant space to it.

However, these clusters did not perform very well beyond 8 or 16 processors and the performance issues got mingled with parallelization strategy, speed of communication, protocol overhead, synchronousness or asynchronousness, the number of cores, etc. The scalability analysis of parallel algorithms and architectures has been a subject of intense discussions [19]. But the fact remains that every problem has its own characteristic complexity, so there is a need to frame a problem which will reflect the structural features that are simple enough and yet provide relative merits of various architectures.

The bandwidth analysis of VARSHA code running on Flosolver MK8 is discussed in [20] where Fig 1 and Table 1 show the dismal performance in the efficiency of VARSHA.

TABLE I. SPEED UP OBTAINED FOR DIFFERENT NUMBER OF BOARDS

Boards	**CPU Processing Time (msec)**	**Communication Time (msec)**	**Actual Processing Time (msec)**	**Speed up**
1	3077	2	3079	1
2	1541	181	1722	1.8
4	774	352	1126	2.7
8	392	528	920	3.3
16	201	700	901	3.4

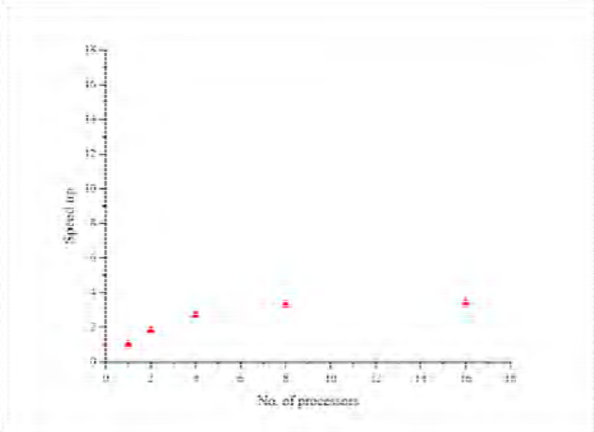

Figure 1. Efficiency of VARSHA code for different no. of processors. (using MPI communication protocol, Ethernet 1GB rating)

It is also discussed that the scaling of bandwidth gives an improved speed up as shown in Fig. 2.

It would be highly appropriate at this point to examine the communication time in such large scale computing systems so that the significance of scaling the bandwidth is clearly perceptible. In this paper we discuss the comparison of various communication devices in cluster based systems for large scale computing environments.

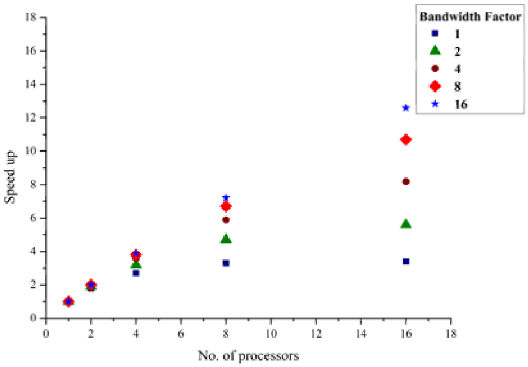

Figure 2. Effect of Bandwidth scaling on speedup.

III. Problem Definition

The following features were used in defining a prototype problem for investigation:

1. In most of the application codes in practice, the communication and computation are disjoint, so that the reference problem should retain this feature.
2. Load balancing is an essential feature of parallel computing, so the reference problem should have ideal load balancing. Furthermore, the reference problem must focus on a hard problem that involves global coupling.
3. The reference problem should reflect global coupling. It is in this class of problems that urgently need parallel computing. This class of problems includes molecular modelling, meteorological computing, ocean modelling, direct simulation of Navier Stokes equations etc.

The above mentioned three guidelines simplify the analysis considerably. Only two parameters then enter the structure; the first is the computation time per step, and the second is the communication time per step. It is not difficult to visualize that if the computation time per step is significantly better or larger than the communication time, scalability follows. Only when the two are comparable does the problem get interesting, and issues of architecture, bandwidth, connectivity, algorithms, decomposition of target applications, etc., arises. In this class, the extreme case is that of a perfectly balanced computational load with no overlap in computing, communication and globally-coupled communication time scales. The performance of parallel computing in various systems, including cluster computing, will provide the necessary parameters to point out where cluster computing stands.

This leads to a further simplification. As the load is perfectly balanced, the subjectivity of specific problems disappear and only a token problem of insignificant computational resource demand can be taken, so that the comparative studies do not involve any application-specific consideration.

IV. Methodology and Observations

Three types of parallel computing platforms have been chosen for the present investigation. The first is the cluster type. The name gives the impression that there are only one to two CPUs per board. For definitiveness, only a single CPU per board is chosen for the experiment, though most of the boards commonly offer two or more processors and many cores per processor. This platform uses Ethernet connectivity for interconnection of these CPU boards. The second platform is the other extreme type having industry standard interconnection for larger number of boards using InfiniBand. And then the platform consisting of Floswitch type of architecture that uses proprietary interconnection scheme. Floswitch type of architecture was developed around the year 2000 at National Aerospace Laboratories, Bangalore. It has the additional feature of message processing. In other words, the communication switch has built in processing ability [21].

A. Sequential Timings

The computational problem has been kept at its simplest. The sequential timings for adding 2 arrays, 4 arrays etc. up to 64 arrays has been obtained for the various configurations. The code is common and given below.

```
/*Initialise*/
int array_length = 1024;
for(i=0;i<array_length;i++)
{
array_1[i] = 1.0;
array_2[i] = 1.0;
.
.
array_n[i] = 1.0;
}
/*Summation Loop*/
for(i=0;i<array_length;i++)
{
sum[i] = array_1[i] + array_2[i]+.....+ array_n[i];
}
```

As expected, the computation time even for more number of processors is insignificant for this experiment and Table 4 contains the sequential computing time.

TABLE II. TIMINGS (IN µSECS) FOR SEQUENTIAL COMPUTATIONS

Configuration type	**No. of arrays**					
	2	*4*	*8*	*16*	*32*	*64*
Ethernet based Cluster type Architecture case	20	22	26	34	50	98
Floswitch based Architecture case	20	22	26	34	50	98
InfiniBand based Architecture case	4	6	10	18	28	58

B. Parallel Timings

For the parallel case, the arrays are added as shown in the following pseudo-algorithm:

```
/*Initialise*/
#define PACKET_COUNT 1
int array_length = 1024;
for(i=0;i<array_length;i++)
{
array[i] = 1.0;
  }
/*Summation across the boards*/
for(i=0;i<PACKET_COUNT;i++)
{
MPI_Allreduce(array,sum,array_length, MPI_FLOAT, MPI_SUM, MPI_COMM_WORLD);
}
```

The details of the interconnects used for parallel computations in the various architecture configurations are given below:

1. *Ethernet based Cluster type configuration*: It employs interconnection using D-Link Gigabit Switch. This Ethernet connectivity has a data rate of 1 Gbps.
2. *InfiniBand based Cluster type configuration*: In this configuration utilises 4x QDR InfiniBand based interconnection having a signalling rate of 40 Gbps is used. Reference [16] has details of acronyms and the related protocols.
3. *Floswitch based Cluster type configuration*: In this configuration, the interconnection is based on the customised communication switch called Floswitch [21]. The Floswitch has communication at dual levels. The PCI interconnect is used for interconnection of 4 boards with a speed of 0.528 GBytes/s that constitutes intracluster communication, and optical interconnects with the data rate of 6.25 Gbps for intercluster communication are used.

The timings obtained for various numbers of processors are as shown below in Table 5. Fig. 3 shows the graph for the communication overhead for various cluster architectures. It may be mentioned that these timings have been obtained over a large sample and in fact, the number of runs made have been over 100 for each case and the variations have been inconsequential.

TABLE III. TIMINGS (IN *M*SECS) FOR PARALLEL COMPUTATIONS ON VARIOUS NO. OF PROCESSORS WITH DIFFERENT ARCHITECTURES

Interconnection type used	**No. of processors**					
	2	**4**	**8**	**16**	**32**	**64**
Ethernet	0.180	0.324	1.034	1.972	3.668	5.467
InfiniBand	0.022	0.032	0.042	0.055	0.066	0.082
Floswitch	0.346	0.572	0.878	1.105	1.495	2.092

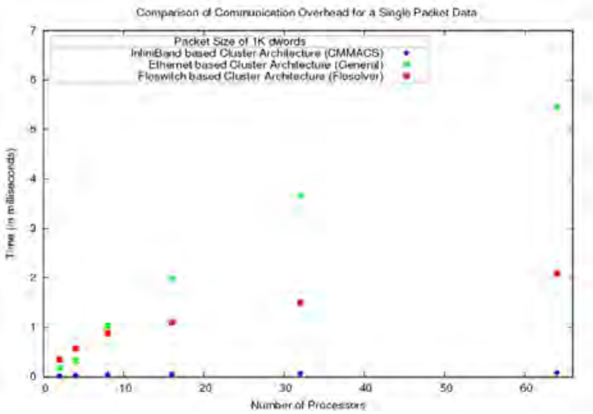

Figure 3. Communication overhead comparisons for various cluster architectures

V. CONCLUSION

It is very clearly visible that in a cluster based architecture, communication time increases with the number of processors for a globally coupled class of problem. Even as InfiniBand based communication retains its supremacy, the Floswitch based communication has its own prominence of low development cost as compared to the InfiniBand. It may be highly agreeable to say that the Floswitch being the novel perception for communication in large scale computing systems, does exhibit appreciable performance.

ACKNOWLEDGMENT

The staff at Flosolver, CSIR-National Aerospace Laboratories, Bangalore provided necessary assistance in carrying out the measurements. We thank them for that. Prof. P Seshu Scientist-in-charge C-MMACS and Dr. G N Dayananda Head CSMST NAL deserve our gratitude for their constant support.

REFERENCES

[1] Mark Delderfield, Lee Kitching, Gareth Smith, David Hoyle and Iain Buchan, “Shared Genomics: Accessible High Performance Computing for Genomic Medical Research”, in Proceedings of the 2008 Fourth IEEE International Conference on eScience (IEEE Computer Society, Washington DC, USA) 404-405 December 2008.

[2] Herman Goldstine H and John Von Nuemann, “Blast Wave Calculation, in John Von Neumann collected works Theory of games”, Astrophysics, Hydrodynamics and Meteorology, Article 29, vol VI, edited by A. H. Taub, (Oxford, Pergamon Press Ltd.) 1976, pp.386 – 412.

[3] Hausheer F H, “Numerical simulation, parallel clusters, and the design of novel pharmaceutical agents for cancer treatment”, in Proceedings of the 1992 ACM/IEEE conference on Supercomputing, edited by Robert Werner, (IEEE Computer Society Press, Los Alamitos, CA, USA) 1992, pp. 636-637.

[4] E. Fermi, J R Pasta, M Tsingou and S Ulam, “Studies of non- linear problems I”, Technical Report LA-1940 (Los Alamos Scientific Laboratory, Los Alamos, NM, USA) 1955.

[5] Jon Squire S and Sandra Palais M, “Programming and design considerations of a highly parallel computer”, in Proceedings of the spring joint computer conference (ACM, New York, NY, USA) 395-400 May 1963.

[6] Koczela L J and Wang G Y, “The Design of a Highly Parallel Computer Organization”, IEEE Trans. Computers, 18, 1969, pp. 520-529.

[7] Akira Kasahara, “Computer Simulations of the Global Circulation of the Earths Atmosphere”, in Computer and their role in the physical sciences, Chapter 23, edited by S Fernbach and A Taub, (Gordon and Breach Science Publishers, New York) 1970, pp. 571-594.

[8] Herman Goldstine H, “The Computer: from Pascal to Von Neumann”, 2nd edn., (Princeton University Press, New Jersey) 1973.

[9] Charles Eames andRay Eames, “A Computer Perspective”, edited by Glen Fleck, (Harvard University Press, Cambridge, Massachusetts) 1973.

[10] David J Kuck, “The structure of Computers and Computation”, Vol 1, (John Wiley & Sons Inc., New York) 1978

[11] Presper Eckart Jr J, “The ENIAC”, in A History of Computing in the Twentieth Century, edited by N Metropolis, J Howlett and Gian Carlo Rotta, (Academic Press, New York) 1980, pp.525 – 540.

[12] John Mauchly W, “The ENIAC”, in A History of Computing in the Twentieth Century, edited by N Metropolis, J Howlett and Gian Carlo Rotta, (Academic Press, New York) 1980, pp. 541 – 550.

[13] Barry Smith F, Petter Bjorstad and William Gropp, “Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations”, (Cambridge University Press) 1996.

[14] F Wolf and B Mohr, “Automatic performance analysis of hybrid MPI/OpenMP applications”, *Proc. of Euromicro Conference on Parallel, Distributed and Network based Processing*, (Genova, Italy) 2003, 13-22.

[15] A S. Tanenbaum, “*Computer Networks*”, **4th edn,** (Pearson Education Inc.) 2007.

[16] T Shanely, “*InfiniBand Network Architecture*”, (Mindshare Inc.) 2003.

[17] “*Interconnection Networks for High – Performance Parallel Computers*”, edited by I D Scherson and A S Youssef (IEEE Computer Society Press) 1994.

[18] Mohammed Mahfooz Sheikh, A M Khan, T Venkatesh and U N Sinha, “The Assessment of Bandwidth Requirements for Meteorological Code VARSHA on a Parallel Computing System”, International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 4, July-August 2012, pp.1261-1268.

[19] T L Sterling, J Salmon, D J Becker and D F Savarese, “*How to Build a Beowulf: A Guide to the Implementation and Application of PC Clusters*”, **2nd edn** (The MIT Press) 1999.

[20] M De, S De and A B Bhattacharya, “Parallel Architecture and algorithms for Space Weather Prediction – A Review”, *Indian Journal of Radio and Space Physics*, **37** (2008) 157-173.

[21] T N Venkatesh, U N Sinha and R S Nanjundiah, “Building a Scalable Parallel Architecture for Spectral GCMS”, *Developments in Terracomputing – Proc. of ninth ECMWF workshop on the Use of High Performance Computing in Meteorology*, (World Scientific) 2001, 1-11.

AUTHORS PROFILE

Mohammed Mahfooz Sheikh obtained his MSc in Electronics from Mangalore University. He is presently pursuing his PhD studies from the Dept of Electronics, Mangalore University under the supervision of Dr. A M Khan. He has also been working on the communication in scientific parallel computing under the guidance of Dr. U. N. Sinha at Flosolver lab, National Aerospace Laboratories, Bangalore.

A M Khan obtained his M.Sc in Applied Electronics from Gulbarga University and his PhD from Mangalore University. He is presently Associate Professor and Chairman, Department of Electronics, Mangalore University. He has around 18 years of teaching and research experience that includes embedded systems, biomedical signal processing and image processing.

U N Sinha obtained his PhD degree from IIT, Kanpur in 1976. He is presently working as Distinguished Scientist at CSIR – National Aerospace Laboratories and CSIR – Centre for Mathematical Modeling and Computer Simulations, Bangalore. He has the credit of building the first parallel computer in India – Flosolver Mk1 in 1986. Dr U N Sinha is India’s pioneer and indisputably the technology leader in the area of parallel computing. His expertise spans over parallel computer hardware design and development, it’s switching technologies, large complex code development, computational fluid dynamics, and numerical weather prediction.