Bandwidth Requirements of Large Scale Computing Systems – A Case Study

A M Khan
Department of Electronics Mangalore University
Mangalagangotri. 574 199

Mohammed Mahfooz Sheikh
Department of Electronics Mangalore University
Mangalagangotri. 574 199

ABSTRACT
Complex scientific problems like weather forecasting, computational fluid and combustion dynamics, computational drug design etc. essentially require large scale computational resources in order to obtain solution to the equations governing them. These solutions can be obtained by developing large legacy codes and then executing them using parallel processing. The parallel processing computers generally demand huge bandwidth as they consist of large number of networked processing elements. One such legacy code VARSHA is a meteorological code used for weather forecasting developed at Flosolver, CSIR-NAL under the joint project from NMITLI (New Millennium Indian Technological Leadership Initiative) and MoES (Ministry of Earth Science). The parallel efficiency of VARSHA code using ethernet connectivity has been anything but satisfactory. This paper discusses the bandwidth utilisation of VARSHA code in its existing and modified forms in order to draw some important conclusions on the bandwidth requirements of the future state-of-art parallel computers used to execute such legacy codes.

General Terms
Bandwidth Scaling, Timing Analysis, VARSHA Code.

Keywords
Bandwidth, computation, communication, speed up.

1. 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 1960's [5,6]. The story of developments of the computers in use till early 70's is well documented and vividly presented in the references [7,8,9,10,11,12].

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 Green's 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).

2. ESSENCE OF COMMUNICATION TECHNOLOGY IN PARALLEL PROCESSING
Communication component being a critical part in building a parallel computer, the bandwidth estimation of a parallel processing system [14] is always a prerequisite and essentially the bandwidth requirement of the application is required to be well within the system bandwidth specifications [15]. Many real time applications like meteorological computing, DNS computing, panel techniques for aircraft wing load calculation and many other problems of this class essentially require parallel architectures for their solutions. The demonstration of super linear speed up of Navier Stokes calculation [16] which is something like a milestone in judging effectiveness of parallel computing is indeed an important initiative .

Large legacy codes that demand global coupling essentially require high speed communications. However it is possible to increase the speed of computations by using more number of PEs where more jobs can be executed in parallel but appropriate communication mechanisms are to be used depending on the intensity of communication demanded by the application. One such parallelised version of legacy code, VARSHA [17] operational at Flosolver lab in NAL is presented in this paper along with the analysis techniques for its bandwidth requirements. Although VARSHA requires large communication bandwidth, initially here cluster based architecture is used for its analysis due to ease of its availability. As the case study is based on VARSHA model, it is briefly described in section 3.

3. VARSHA MODEL

This VARSHA model is a global/general circulation model of atmosphere that solves a set of nonlinear partial differential equations to predict the future state of the atmosphere from a given initial state. Since the domain of atmospheric flow is bounded at the bottom by the surface of the earth, exchange of properties take place at this surface and it is necessary to prescribe appropriate boundary conditions or values for various quantities. The bottom topography plays an important plays an important role in controlling the airflow not only close to the ground but also at upper levels through induced vertical motion and momentum transfer by gravity waves. Present day atmospheric models have moisture as one of the variables and take into account diabatic processes like evaporation and condensation. All physical processes involving moisture and others like radiation, turbulence, gravity wave drag, land surface processes etc. are parameterized in terms of variables in the VARSHA model. Detailed discussion can be found in [18,19] and the details of parallelisation are available in [17].

4. BANDWIDTH UTILISATION OF VARSHA MODEL

Large application codes developed till late 70's or early 80's abound; these codes are developed around sequential computer having von Neumann's architecture. Extension of these codes both in terms of scope and efficiency is a natural requirement which can only occur through the only possible route i.e. parallel routes. Such codes are commonly called as legacy code. Code VARSHA used for weather predictions, mainly consists of numerical computation of equations in the spectral domain and communication of data at each time step of forecast that demands higher bandwidth. The parallel simulation was done in Flosolver MK3 parallel computer at NAL and [15] contains the data on its performance. The key point in this simulation was that a GCM model could be run on 4 processor Flosolver MK3 which was a remarkable feet and the efficiency issues were related to a second place as platforms having large number of processors were not available at that time. In 2010, the pictures have changed, the numbers of processors available are large and the issue is now that of parallel efficiency. Practically, the same VARSHA code running on Flosolver MK8 (1024 Xeon processors @ 2GHz and 4TB RAM) gives the efficiency shown in the Figure 1 which is dismally poor.

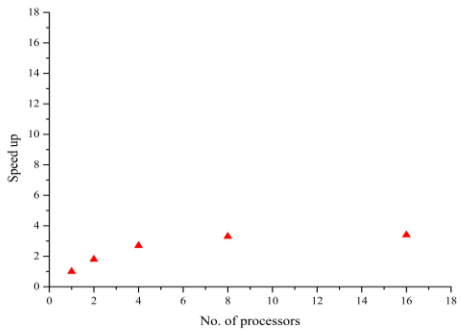


Figure 1. Efficiency of VARSHA code for different no. of processors. (using MPI communication protocol, Ethernet 1GB rating)

These computations in the fourth time steps and their communication to other boards depending on the number of boards are profiled and presented in Table 1. The speedup trend is as shown in the previous graph in Figure 1. Rewriting of this code in order to utilise the time utilisation of compute intensive part for communication of already computed data values through an external port shall reduce the effective execution time of the code which is explained in the following section.

Table 1. Speedup obtained for original application code

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

5. BANDWIDTH REQUIREMENTS OF MODIFIED VARSHA CODE

The profiled timings for computation and communication for different number of processors is as shown in Table 1. It is evident from Table 1 that the speedup is not appreciable beyond 8 processors. But if the bandwidth can be improved, the communication time will be reduced and there will be increase in speedup for more number of processors. In other words, if t is the actual processing time then

$$t = t_{cpu} + t_{comm} \qquad \text{.....(1)}$$

where t_{comp} is the computation time and t_{comm} is the communication time. The bandwidth being increased by a factor of k, the effective processing time then computed would be

$$t = t_{cpu} + \frac{t_{comm}}{k} \text{ ; where } k \text{ is a positive integer. .....(2)}$$

The scaling of bandwidth is utmost vital in deciding the computational efficiency of the legacy code. The effect of bandwidth scaling and the overlapping techniques were carried out using profiling tools built in house. However, as the emphasis is on bandwidth scaling and overlapping techniques for the legacy code, the profiling techniques shall be discussed elsewhere.

5.1 Effect of Bandwidth scaling

Let us consider an example from Table 1, for an 8 processor system, if the bandwidth is increased by a factor of 8 (i.e. k = 8); CPU processing time will remain the same, i.e. 392msec but communication time will be $\frac{528}{8} = 66$ msec, and the speedup will become 6.7. In fact, the Table 1 will be modified for k = 8 in equation (2) as shown in Table 2.

Similarly, for a 16 processor system; CPU processing time is 201msec and communication time will be $\frac{700}{8} = 87.5$ msec, then the speedup will become 10.7. Therefore, in an 8 processor system, the efficiency is increased from 3.3 to 6.7 and in a 16 processor system the efficiency is increased from 3.4 to 10.7. The Table 3 gives the speedup values for different bandwidth of the order $k = 2^m$ where $0 \le m \le 4$.

Table 2: Speedup obtained for original application code

Boards	CPU Processing Time (msec)	Communication Time (msec)	Actual Processing Time (msec)	Speed up
1	3077	0.3	3077.3	1
2	1541	22.6	1563.6	2
4	774	44.0	818.0	3.8
8	392	66.0	458.0	6.7
16	201	87.5	288.5	10.7

Table 3. Bandwidthwise Speedup of original test code

Scaling of Bandwidth (k)	No. of Boards				
	1	2	4	8	16
1	1.0	1.8	2.7	3.3	3.4
2	1.0	1.9	3.2	4.7	5.6
4	1.0	1.9	3.6	5.9	8.2
8	1.0	2.0	3.8	**6.7**	**10.7**
16	1.0	2.0	3.9	7.2	12.6

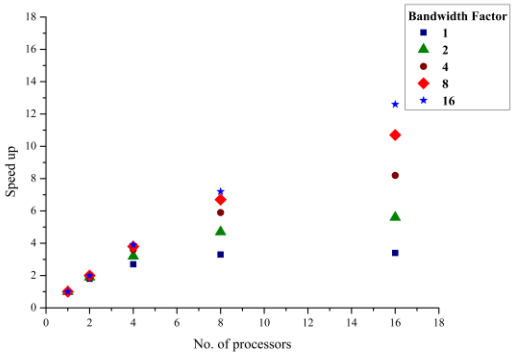


Figure 2. Effect of Bandwidth scaling on speedup.

The graph in Figure 2 shows the change in speedup for various bandwidths. It is observed that as the bandwidth scaling increases i.e. the communication time reduces, the speedup of the application code increases when more no. of processors are used.

5.2 Effect of Bandwidth Scaling on Modified Code

For better parallelisation efficiency, the option of code rearrangement plays a vital role-very often not considered seriously while handling legacy codes. For example, in the present code for which data is presented in Table 1, the operations of computation and communication are disjoint; but in the case if the code is rewritten the computation and communication may be made to overlap, one will have the following table of efficiency as shown in Table 5.

Then effective processing time in equation (1) would be

$$t = t_{cpu} + t_{comm} - t_{ov} \quad \text{.....(3)}$$

where t_{cpu} is the CPU processing time, t_{comm} is the communication time and t_{ov} is the overlapped communication time. Then again in Table 1, considering the case of 8 boards, the CPU processing time is 392 msec and communication time is 528msec, the overlapped communication time is 392 msec, then the effective processing time will be 392+528-392=528 msec and the efficiency is $\frac{3077}{528} = 5.8$ as shown in Table 4.

Table 4. Speed up obtained for modified code with different number of boards

Boards	CPU Processing Time (msec)	Communication Time (msec)	Overlapped communication time (msec)	Effective processing time (msec)	Speed up
1	3077	2	2	3077	1
2	1541	181	181	1541	2.0
4	774	352	352	774	4.0
8	**392**	**528**	**392**	**528**	**5.8**
16	201	700	201	700	4.4

The graph in Figure 3 shows the trend in the speedup for different number of processors. If the bandwidth scaling is observed in the case of modified code, the bandwidth being increased by a factor of k, the calculation of effective processing time in equation (2) becomes,

$$t = t_{cpu} + \frac{t_{comm}}{k} - t_{ov} \quad \text{.....(4)}$$

The comparative figures for the different bandwidth scaling of application code and modified application codes are given in Table 5. Figure 4 shows the comparison graph for various bandwidths scaling for original and modified codes. It is observed that the performance speedup in the case of modified code is dramatic for sizeable scaling of bandwidth.

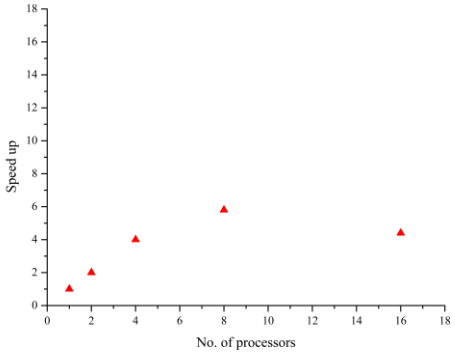
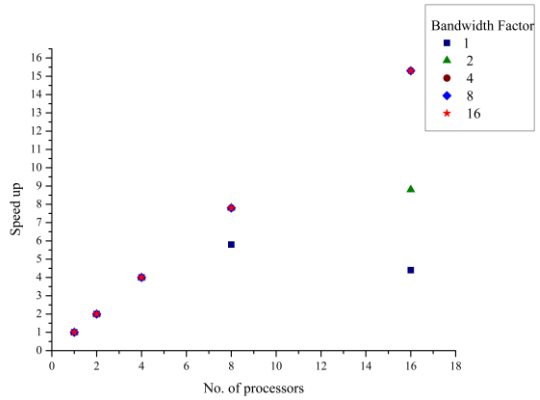


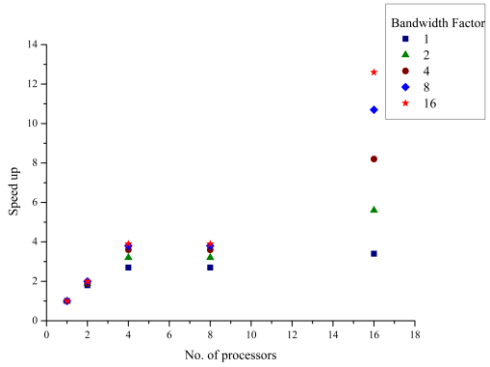
Figure 3. Speedup in case of modified code

Table 5. Effect of Bandwidth on original and modified code (small parallel processing)

Scaling of Bandwidth (k)	No. of boards					Remarks
	1	2	4	8	16	
1	1.0	1.8	2.7	2.7	3.4	Old
	1.0	2.0	4.0	5.8	4.4	Modified
2	1.0	1.9	3.2	3.2	5.6	Old
	1.0	2.0	4.0	7.8	8.8	Modified
4	1.0	1.9	3.6	3.6	8.2	Old
	1.0	2.0	4.0	7.8	15.3	Modified
8	1.0	2.0	3.8	3.8	10.7	Old
	1.0	2.0	4.0	7.8	15.3	Modified
16	1.0	2.0	3.9	3.9	12.6	Old
	1.0	2.0	4.0	7.8	15.3	Modified



(a)



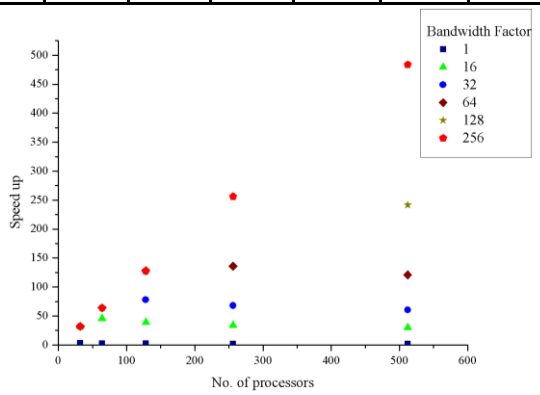
(b)

Figure 4. Comparison of Speedup between original and modified code for varying Bandwidth

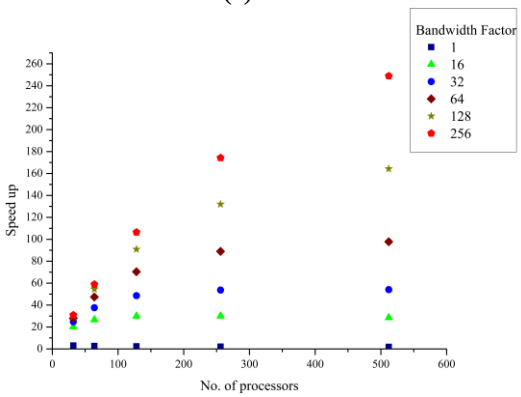
It will be interesting to have a comparison table for large number of processors and scaling of bandwidth as shown in Table 6, so that performance issues may be put into perspective. Figure 5 shows the comparison graph point for various higher bandwidths scaling of the order up to 512, for original and modified codes. It shows that as the no. of processors increases, for the bandwidth scaling by a considerable scale (say half the no. of processors), the performance speedup in the case of modified code is almost double.

Table 6. Effect of Bandwidth on original and modified code (moderate parallel processing)

Scaling of Bandwidth (k)	No. of boards					Remarks
	32	64	128	256	512	
1	3.1	2.7	2.4	2.1	1.9	Old
	3.4	2.8	2.4	2.1	1.9	Modified
16	20.2	26.6	29.9	30.0	28.5	Old
	32.1	45.6	39.1	34.0	30.2	Modified
32	24.8	37.6	48.6	53.7	54.1	Old
	32.1	64.1	78.1	68.0	60.5	Modified
64	28	47.4	70.4	89.0	97.8	Old
	32.1	64.1	128.2	136.0	120.9	Modified
128	29.9	54.5	90.9	132.0	164.3	Old
	32.1	64.1	128.2	256.4	241.7	Modified
256	30.9	58.9	106.4	174.3	248.9	Old
	32.1	64.1	128.2	256.4	483.8	Modified



(a)



(b)

Figure 5. Comparison of Speedup between original test code and modified test code for higher Bandwidth

6. CONCLUSION

The VARSHA code has been analysed for its bandwidth requirements for both its existing and modified forms. It is noted that, if the overall processing time (Table 1) has to be

decreased from 3079 msec to around 800 msec i.e. processing time be reduced roughly by a factor of 4, it is not enough to increase the number of boards from 1 to 16 which is far more than 4 needed for expected reduction; instead if the number of boards are only increased from 1 to 4 and the bandwidth increased from 1 to 8 (Table 2), the timing requirements will be met. This clearly brings out the fact that by increasing number of boards or CPUs or cores is insufficient for reducing processing time, it needs to be backed up by the increase in communication bandwidth.

The analysis suggests that a present day parallel processing centre needs to have network hardware which supports bandwidth on demand keeping overall resources intact. It is understandable that at a given point of time all the tasks will not require peak bandwidth; hence it is meaningful to consider that the resources can be combined. Currently such hardware do not exist, suggesting that there is a need to develop such class of hardware if these centres are not identified with specific application programmes. This idea shall give rise to a completely new paradigm of bandwidth on demand in parallel computing.

7. ACKNOWLEDGMENTS

Author is extremely thankful to the Director, NAL & Dr. U N Sinha, Distinguished Scientist, NAL/4PI, Bangalore for their unwavering support.

8. REFERENCES

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

[2] Herman Goldstine H & John Von Nuemann. 1976. Blast Wave Calculation. 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.), 386 – 412.

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

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

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

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

[7] Akira Kasahara. 1970. 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 & A Taub, (Gordon and Breach Science Publishers, New York), 571-594

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

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

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

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

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

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

[14] Shioda, S and Mase, K. 2004 . A new approach to bandwidth requirement estimation and its accuracy verification, IEEE Intl. Conf. on Communications 2004, Vol 4, 1953 – 1957.

[15] Yanping Zhao, Eager, D.L. ; Vernon, M.K. 2007. Network Bandwidth Requirements for Scalable On-Demand Streaming, IEEE/ACM Transactions on Networking, Vol 15,No 4, 878 – 891.

[16] Venkatesh T N, Sarasamma V R., Rajalakshmy S, Kirti Chandra Sahu, Rama Govindarajan. 2005. Super-linear speed-up of a parallel multigrid Navier – Stokes solver on Flosolver, Current Science, 88(4), 589 – 593.

[17] U. N. Sinha, V. R. Sarasamma, S. Rajalakshmy, K. R. Subramanian, P. V. R. Bharadwaj, C. S. Chandrashekar, T.N.Venkatesh, R.Sunder, B.K.Basu, Sulochana Gadgil and A. Raju, *Monsoon Forecasting on Parallel Computers*, Current Science, Vol 67,No 3,178-184, August 1994.

[18] Holton, J. R., and Hsiu-Chi Tan. 1980. The influence of the equatorial quasi-biennial oscillation on the global circulation at 50 mb. J. Atmos. Sci., Vol 37, 2200-2208.

[19] Holton, J. R. and H.-C. Tan. 1982. The quasi-biennial oscillation in the Northern Hemisphere lower stratosphere. J. Meteor. Soc. Japan, Vol 60, 140- 148.