The Assessment of Bandwidth Requirements for Meteorological Code VARSHA on a Parallel Computing System

Mohammed Mahfooz Sheikh*, A M Khan*‡, T Venkatesh and U N Sinha†**

*(Dept. of Electronics, Mangalore University, Mangalore, Karnataka 574199.)
** (Dept. of Computer Science and Engineering, Ghousia College of Engineering, Ramanagaram)
† (Distinguished Scientist, CSIR NAL/C-MMACS, Bangalore, Karnataka 560017)

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 are obtained by developing large legacy codes and then executing them using parallel processing systems as they require large scale computations. 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 VARSHA is being run on a large scale computing platform, the Flosolver Mk8, the latest of the Flosolver series of parallel computers. This paper discusses the bandwidth utilisation of VARSHA code on Ethernet based interconnect, in its existing and modified forms in order to access the bandwidth requirements of such legacy codes on parallel computing systems.

Keywords - **Meteorological code, Bandwidth scaling, Parallel Computing, VARSHA model, Flosolver**

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]. New drug development routinely uses large scale computing [2]. 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 [3]; 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 [4], [5]. The story of developments of the computers in use till early 70s is well documented and vividly presented in the references [6–11]. 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 [12]. 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 [13] is always a prerequisite and essentially the bandwidth requirement of the application is required to be well within the system bandwidth specifications [14]. 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 which is something like a milestone in judging effectiveness of parallel computing [15] 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 [16] operational at Flosolver lab in NAL is presented in this paper along with the assessment for its bandwidth requirements. Although VARSHA requires large communication bandwidth, here cluster based architecture is used for its assessment 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 [17], [18] and the details of parallelisation are available in [16]. The governing equations for the VARSHA model are discussed below.

The model is based on the conversation laws for mass, momentum, energy and moisture. The momentum equations are replaced by vorticity and divergence equations so that the spectral techniques can be applied in the horizontal direction. The vertical coordinate is $\sigma = \frac{p}{p_*}$ where p is the layer pressure and p_* the surface pressure. Finite differences are used to approximate the differential operators in the vertical directions. The governing equations are cast in the form of evolution equations for the model variables viz. temperature(T), surface pressure (p_*), specific humidity(q), divergence(D) and vorticity(η). The model prognostic equations are given below:

The thermodynamic equation is $\frac{\partial}{\partial t}\ln\theta = \frac{H}{C_p T}$

where H is the heating rate per unit mass and θ is the potential temperature. This is rewritten to give the following equation for temperature

$$\frac{\partial T}{\partial t} = -V\cdot\nabla T + kT\left(\frac{\partial}{\partial t} + \vec{V}\cdot\nabla\right)\times\ln p_* + \frac{H}{C_p} - \pi\dot{\sigma}\frac{\partial}{\partial\sigma}\left(\frac{T}{\pi}\right) \quad (1)$$

where T = π θ; $\pi = p_*^k; k = \frac{R}{C_p}$; and ∇ is the horizontal gradient in the system.

The continuity equation is

$$\frac{\partial}{\partial t}\ln p_* + \vec{V}\cdot\nabla\ln p_* + \nabla\cdot\vec{V} + \frac{\partial\dot{\sigma}}{\partial\sigma} = 0 \quad (2)$$

On integrating this equation with the boundary conditions $\dot{\sigma}(0) = \dot{\sigma}(1) = 0$ we get the equation for surface pressure,

$$\int_0^1\left(\frac{\partial}{\partial t}\ln p_*\right)\partial\sigma = -\int\left(\nabla\cdot\vec{V} + \vec{V}\cdot\nabla\ln p_*\right)\partial\sigma$$

(3)

The conversation law for moisture is $\frac{dq}{dt}$ = S, where q is the specific humidity and S represents the sources and sinks. This equation is written as

$$\frac{\partial q}{\partial t} = -\vec{V}\cdot\nabla q - \dot{\sigma}\frac{\partial q}{\partial\sigma} + S \quad (4)$$

The equations for divergence and vorticity are obtained from the momentum equation by taking the dot and cross products respectively. The equation for divergence is

$$\frac{\partial D}{\partial t} = \frac{1}{a\cos^2\phi}\left(\frac{\partial B}{\partial\lambda} - \cos\phi\frac{\partial A}{\partial\phi}\right) - \nabla^2\left(E + \Phi + RT_0\ln p_*\right) \quad (5)$$

and the equation for vorticity is

$$\frac{\partial \eta}{\partial t} = -\frac{1}{a\cos^2\phi}\left(\frac{\partial A}{\partial \lambda} - \cos\phi\frac{\partial B}{\partial \phi}\right) \qquad (6)$$

where D = $\nabla \cdot \vec{V}$; η = $f + \zeta_k$; $\zeta_k = \vec{k}\cdot\nabla\times\vec{V}$

$$A = \eta U + \left(\frac{RT}{a}\right)\cos\phi\left(\frac{\partial}{\partial t}\ln p_*\right) + \dot{\sigma}\frac{\partial V}{\partial \sigma} - F_\phi \cos\phi$$

$$B = \eta V - \left(\frac{RT}{a}\right)\left(\frac{\partial}{\partial \lambda}\ln p_*\right) + \dot{\sigma}\frac{\partial U}{\partial \sigma} - F_\lambda \cos\phi$$

$$U = u\cos\phi; V = v\cos\phi$$

$E = \frac{\vec{V}\cdot\vec{V}}{2}$; Φ = geopotential height; f = Coriolis parameter;
a = radius of the earth; φ = latitude; λ = longitude; Fφ = zonal component of the dissipative process; Fλ = meridional component of the dissipative processes. The VARSHA code has a horizontal resolution of 80 waves in triangular truncation and 18 atmospheric layers in the vertical. The Fourier space again has 80 waves in the EW on 128 quasi-equidistant Gaussian latitudes which exclude both the poles and the equator. In the physical grid space the Fourier waves are projected onto 256 equispaced longitude points on each Gaussian latitude. The number of longitude points has to be much larger than the azimuthal wave number in order to resolve the shorter waves produced by nonlinear interaction. The VARSHA forecast model is fully diabatic and includes parameterisation of all known physical processes like radiation, convection, turbulent boundary layer processes, land air interaction, surface friction, etc.

4. PARALLELISATION STRATEGY 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 model computations are done in three distinct spaces namely the two dimensional global spectral space, the one dimensional Fourier space (spectral in azimuthal direction) for each latitude and the two dimensional latitude/longitude grid space. Since the basic functions in the spectral expansion are orthogonal to each other, all linear operations in the spectral space involve only the selected basic function and such computations can be done in parallel. Parts of the nonlinear terms are computed in the Fourier space, while the rest along with the physical parameterization computations are carried out on the latitude/longitude grid. The natural flow of the model code is to compute the Fourier coefficients for each latitude, complete calculation in the Fourier space and then move to the grid space through the discrete Fast Fourier Transform (FFT). After completion of computations in the grid space, mostly related to physical parameterizations which work independently at each grid point along a vertical column, atmospheric variables are again moved to Fourier space through FFT. Finally, at the end of each time step of integration, full spectral coefficients are again formed by summing up contributions from all the latitudes. Except for this last operation all computations involve quantities specific to one latitude only. Thus, a natural scheme of parallelization considered was to distribute the latitudes to different processors. This essentially implies a domain splitting in the Fourier space only. The latitudes were divided among the processors and that each time step demands global communication in order to sum the full spectral coefficients of each latitude. The parallel simulation was done in Flosolver MK3 parallel computer at NAL and [16] 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. 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 Fig. 1 which is dismally poor.

5. BANDWIDTH ASSESSMENT FOR LEGACY CODE VARSHA

The initial portion and the completion phase of the VARSHA model, the computational load is quite different and is representation of the main computation task. Thus, there is a need to profile only a specific segment of the code which is compute intensive. Moreover, the complete profiling of VARSHA will generate huge amount of data. With all these computations in mind, the present study utilises fourth time step of the forecast for profiling. During each of the time steps, the code computes, Fourier Transform of 256 latitudes of 512 points each for 18 atmospheric levels. It also computes Legendre Transform for equal number of points and also computes the nonlinear part of dynamical calculations. These computations in the fourth time step and their communication to other boards depending on the number of boards are profiled and presented in Table I.

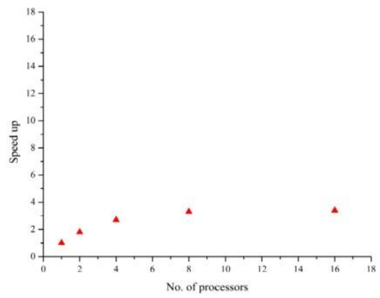


Fig. 1. Efficiency of VARSHA model for various number of processors. (using MPI communication protocol, Ethernet 1GB rating)

TABLE I SPEEDUP OF 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

The speedup trend is as shown in the previous graph in Fig. 1. Also the bandwidth assessment techniques for the speedup trend in Fig. 1 are explained as follows:

5.1 Scaling of Bandwidth

It is evident from Table I 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

$$t = t_{cpu} + t_{comm} \qquad (7)$$

where tcomp is the computation time and tcomm is the communication time. The bandwidth being increased by a factor of k, the effective processing time then computed would be

$$t = t_{cpu} + \frac{t_{comm}}{k} \quad \text{; where k is a positive integer} \qquad (8)$$

Here, as k is inversely proportional to the actual processing time t, any increase in k value shall tend to minimise the time t and increase the speedup as evident from the Table II.

5.2 Overlapping of Communication and Computation Times

The efficiency of parallel computing is governed by single factor synergy between computation and communication. Basic problem remain the same but the way it is partitioned determines the efficiency. It is extremely essential to keep a balance between the computation and communication times. It is easily seen that the rate of communication time to computation times has to be small for parallel computing to be attractive. But its implementation even in the case of VARSHA often needs massive rewriting of the code and the structure of such rewriting gets mapped by the experiments performed using the profiling tools. In practical execution of VARSHA code, one finds that the computation of each physical parameter at each grid point along the vertical coloumn takes place independently. Once the computation for a single time step (i.e. computation of all parameters) is complete, the values are then communicated globally. Such scenarios where communication waits for completion of computation and vice versa, have to be eliminated and a better method for synergy between computation and communication need to be followed. At the end of computation of each physical parameter, the communication can be initiated at the background such that it does not hamper or delay the computation of next parameter. Towards the end, when the computation of all the parameters are complete and at the time when global communication would normally take place, the communication for each parameters has already been completed. Very often the legacy codes are parallelized in simple minded approach where the computation and communication are disjoint and can be overlapped. This overlapping will provide dramatic reduction in overall execution time of the code. Then effective processing time in equation (7) would be

$$t = t_{cpu} + t_{comm} - t_{ov} \qquad (9)$$

where t_{cpu} is the CPU processing time, t_{comm} is the communication time and t_{ov} is the overlapped communication time and is presented in the Table IV. The effect of bandwidth scaling and the overlapping techniques were carried out using profiling tools built in house. The profiling tools can determine the actual computation and communication times during the course of the code execution. However, as the emphasis is on bandwidth scaling and overlapping techniques for the legacy code, the profiling techniques shall not be discussed here. The timing analysis of the VARSHA code for these techniques is explained in the subsequent sections.

TABLE II SPEED UP IN CASE OF BANDWIDTH SCALED BY FACTOR OF 8

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	2.0
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 III SPEEDUP OF ORIGINAL TEST CODE IN TERMS OF BANDWIDTH

Factor of Bandwidth Increase (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

6. TIMING ANALYSIS OF VARSHA CODE

The bandwidth requirements of the VARSHA code shall be assessed using the communication and computation times. The timing analysis using the techniques explained in section 5 is as described below:

6.1 Effect of Bandwidth scaling on VARSHA

Let us consider an example from TABLE I, 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 I will be modified for k = 8 in equation (8) as shown in TABLE II. 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 III gives the speedup values for different bandwidth of the order $k = 2^m$ where $0 \leq m \leq 4$. The graph in Fig. 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 number of processors are used.

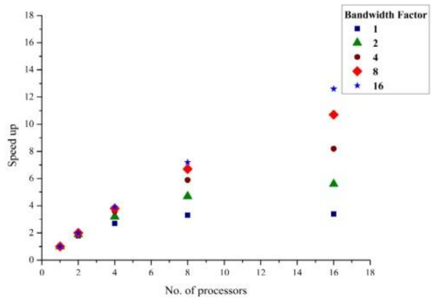


Fig. 2. Effect of Bandwidth scaling on speedup for VARSHA model.

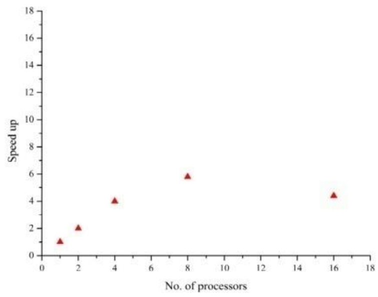


Fig. 3. Speedup in case of modified code.

Thus, the scaling of bandwidth is extremely significant in deciding the computational efficiency of the legacy code.

6.2 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 I, the operations of computation and communication are disjoint; but in the case if the code is rewritten or modified as the computation and communication may be made to overlap, one will have the following table of efficiency as shown in TABLE V. Then again in TABLE I, 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 IV. The graph in Fig. 3 shows the trend in the speedup for different numbers 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 (9) becomes,

$$t = t_{cpu} + \frac{t_{comm}}{k} - t_{ov} \qquad (10)$$

The comparative figures for the different bandwidth scaling of application code and modified application codes are given in TABLE V.

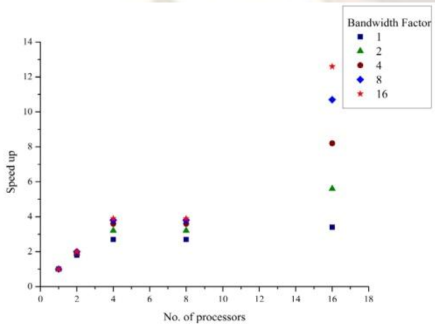
TABLE IV SPEEDUP OF MODIFIED CODE

Boards	CPU Processing Time (msec)	Communication Time (msec)	Communication Time (Overlapping accounted) (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

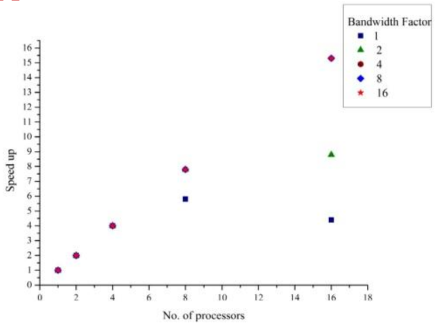
TABLE V SIGNIFICANCE OF BANDWIDTH IN ORIGINAL AND MODIFIED CODES (SMALL PARALLEL PROCESSING)

Factor of Bandwidth Increase (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

Fig. 4 shows the comparison graph for various bandwidth scaling for original and modified codes. It is observed that the performance speedup in the case of modified code is large for sizeable scaling of bandwidth.



(a) Bandwidth inference of original code



(b) Bandwidth inference of modified code

Fig. 4. Comparative significance of Bandwidth in original and modified test codes

It will be interesting to have a comparison table for large number of processors and scaling of bandwidth as shown in TABLE VI, so that performance issues may be put into perspective. Fig. 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 number of processors increases, for the bandwidth scaling by a considerable scale (say half the number of processors), the performance speedup in the case of modified code is almost double.

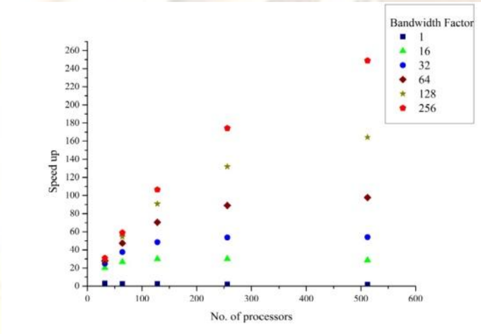
7. CONCLUSION

The VARSHA code has been assessed for its bandwidth requirements for both its existing and modified forms. It is noted that, if the overall processing time (TABLE I) 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, actually required 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 II), the timing requirements shall be very well 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.

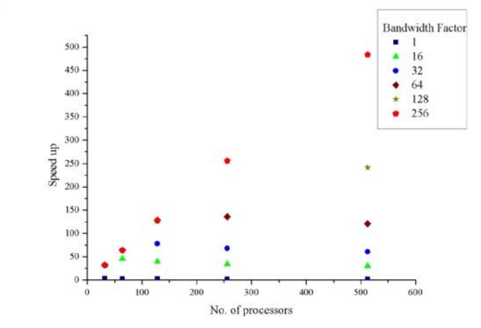
The assessment suggests that a present day parallel processing centre needs to have network hardware which supports bandwidth on demand keeping overall resources intact. It is understandable that at a given point of time all the tasks will not require peak bandwidth. Hence it is meaningful to consider 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 programs. This idea shall give rise to a completely new paradigm of bandwidth on demand in parallel computing.

TABLE VI SIGNIFICANCE OF BANDWIDTH IN ORIGINAL AND MODIFIED CODES (MODERATE PARALLEL PROCESSING)

Factor of Bandwidth Increase(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) Bandwidth inference of original code



(b) Bandwidth inference of modified code
Fig. 5. Comparative significance of higher Bandwidth in original and modified test codes

8. ACKNOWLEDGEMENTS

One of the authors Mohammed Mahfooz Sheikh would like to acknowledge CSIR for the support in the form of fellowship as CSIR-SRF vide CSIR award No. 31/3(38)/2009-EMR-I. Authors are extremely thankful to the Director, NAL, Head, Flosolver Unit and all the scientists at Flosolver for their unwavering support in carrying out this work.

REFERENCES

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

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

[3] Fermi E, Pasta J R, Tsingou M and Ulam S, Studies of nonlinear problems I, *Technical Report LA1940* (Los Alamos Scientific Laboratory, Los Alamos, NM, USA) 1955.

[4] Jon Squire S and Sandra Palais M, Programming and design considerations of a highly parallel computer, in *Proc. of the spring joint computer conference*, ACM, New York, USA, 1963, 395-400.

[5] Koczela L J and Wang G Y, The Design of a Highly Parallel Computer Organization, *IEEE Trans. Computers, 18*, 1969 520529.

[6] Akira Kasahara, Computer Simulations of the Global Circulation of the Earths Atmosphere, in S Fernbach and A Taub (Eds.), *Computer and their role in the physical sciences, Chapter 23*, (Gordon and Breach Science Publishers, New York) 1970, 571594.

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

[8] Charles Eames and Ray Eames, *A Computer Perspective*, Glen Fleck (Ed.), (Harvard University Press, Cambridge, Massachusetts) 1973.

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

[10] Presper Eckart Jr J, *The ENIAC, in A History of Computing in the Twentieth Century*, N Metropolis, J Howlett and Gian Carlo Rotta (Eds.), (Academic Press, New York) 1980 525-540.

[11] John Mauchly W, *The ENIAC, in A History of Computing in the Twentieth Century*, N Metropolis, J Howlett and Gian Carlo Rotta (Eds.),, (Academic Press, New York) 1980 541 550.

[12] Barry Smith F, Petter Bjorstad and William Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, (Cambridge University

Press) 1996.

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

[14] Yanping Zhao, Eager, D.L. ; Vernon, M.K. , Network Bandwidth Requirements for Scalable OnDemand Streaming, *IEEE/ACM Transactions on Networking, 15(4)*, 2007, 878-891.

[15] T N Venkatesh, V R Sarasamma, S Rajalakshmy, Kirti Chandra Sahu, Rama Govindarajan, Superlinear speedup of a parallel multigrid Navier Stokes solver on Flosolver, *Current Science, 88(4)*, 2005, 589-593.

[16] 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, 67(3), 1994, 178-184.

[17] Holton, J. R., and HsiuChi Tan, The influence of the equatorial quasibiennial oscillation on the global circulation at 50 mb. *J. Atmos. Sci., 37*, 1980, 2200-2208.

[18] Holton, J. R. and H.C. Tan, The quasibiennial oscillation in the Northern Hemisphere lower stratosphere. *J. Meteor. Soc. Japan, 60*, 1982, 140-148.