

# **Performance Comparison of Steady State Heat Conduction using MPI, OpenMP and Hybrid MPI-OpenMP**

Anvitha, Pallavi K N

PG Student, Department of Computer Science and Engineering, NMAMIT, Nitte, Karkala, Udupi, Karnataka ,India,

Affiliated to VTU -Belgaum

Assistant Professor, Department of Computer Science and Engineering, NMAMIT, Nitte, Karkala, Udupi, Karnataka

IndiaAffiliated to VTU -Belgaum

**ABSTRACT:** Among many numerical techniques that are used for solving steady state heat conduction problem, Finite Difference Method (FDM) technique has been chosen, and a three dimensional code is developed. A cube is chosen for analysing, and temperature at each discrete point of the cube is computed using the developed code. It is observed that, parallelization of the code using hybrid Message Passing Interface (MPI) and Open Multi Processing (OpenMP) is more efficient for larger grids than using MPI or OpenMP alone. Hence hybrid OpenMP-MPI approach is more desirable to choose because of its cost effectiveness and massive computational speed.

**KEYWORDS:** FDM, MPI, OpenMP

## **I. INTRODUCTION**

Over the past few decades, there have been a lot of researches and improvements done in the field of parallel computation. Researchers have been working hard for optimizing many algorithms and hence improving their performance. Parallelizing the code is the more desirable method to improve the performance of a program or an application. But all the problems cannot be benefitted from parallelization. Problems which are very large and do not have dependencies are highly benefitted from parallelization. Steady state heat conduction is one of such problem.

To study the behaviour of thermal materials, simulation of heat transfer phenomena is very essential. Heat transfer phenomena include heat conduction, heat convection and heat radiation, among which we have chosen heat conduction for our experiment. Whenever there exists a temperature difference within the body of a conduction material, heat conduction occurs; specifically passing energy from molecule to molecule of a material. Heat conduction is of two types: one is steady state heat conduction and another one is unsteady state heat conduction. In this paper we focus only on steady state heat conduction, as it has many applications in the field of engineering. Since lots of computation is involved in discretization and calculating temperature at each point, this is considered to be a large and time consuming problem.

Many of the heat conduction problems can be formulated using Partial Differential Equation (PDE). Then analytical or numerical methods are applied to PDEs to obtain the solution. Analytical methods are suitable only for simple problems and can easily be obtained. For more complex problems, numerical methods such as Finite Difference Method (FDM), Finite Volume Method (FVM) and Finite Element Method (FEM) are more appropriate. Heavy computational work is needed to solve a PDE using numerical methods.

Shivanandan et al [1] mathematically modelled the steady state heat conduction using Laplacian equation [x]. They used FDM method to solve the equation and developed an algorithm for the obtained solution. Even though the

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

algorithm solves steady state heat conduction problem, improving the performance of the algorithm is a very important task. Hence they parallelized the code using Open Multi Processing, Message Passing Interface, and Compute Unified Device Language. In this paper, we implemented the same algorithm using Hybrid OpenMP-MPI approach and found that to achieve substantial improvement in the computational speed, Hybrid OpenMP-MPI approach is very useful than using MPI or OpenMP alone .

Parallel Programming is the vast subject where many approaches such as OpenMP, MPI etc. are available to parallelize the program. Selecting the best approach for the given application or for program is really a challenging task. Hybrid parallel programming is an emerging approach of parallelization and is more useful for the effective utilization of available resources in modern supercomputers. Hybrid approach can be applied in Symmetric Multiprocessing (SMP) clusters, where OpenMP is used to exploit loop level parallelism, and MPI is used for communication between the nodes.

## II. RELATED WORKS

Researchers have been working to parallelize the various numerical solutions of PDEs. Sonia Rani et al [2] developed an application by implementing FEM and FDM on a cluster of message passing. They found that, by using MPI and OpenMP frameworks, 2x to 3x speed-ups can be achieved for sufficiently large sized meshes. In the past years, various methods have been proposed to solve steady state heat conduction equation. Yan et al [3] showed, how a meshless method based on least squares discretization can be used to solve steady state heat conduction equation. They showed that, their MWLS method is an elegant method if both accuracy and efficiency are taken into consideration.

Many researchers have been working on SMP clusters and hybrid parallel programming approach for solving many time consuming problems. Drosinos et al [4] compared the performance of hybrid OpenMP-MPI on SMP cluster for some nested loop algorithms, and found that hybrid approach performs well in some cases compared to pure MPI and pure OpenMP. They tested the overall performance of algorithm on 4-node and 2-node clusters separately. Jones et al [5] also worked on the same concept and compared the performance of 3D-Ordered-Subsets-Expectation-Maximization (OSEM) algorithm on pure MPI, OpenMP and hybrid OpenMP-MPI approach. Both of the above mentioned papers show the importance of hybrid approach in cluster computing. From these papers what we understood is hybrid approach suits well if need of communication between the processors is less compared to time used for computation. Diaz et al [6], Mehrothra et al [7] clearly given the reason why we should go for hybrid approach. They also provided the overall view of many of the parallel programming paradigms. They presented the features, advantages and draw backs of all the languages which support parallelism. Tsuji et al [8] revealed that, for single core multiprocessors, hybrid OpenMP-MPI does not result in good performance, but it performs well in the multicore processors.

Shivanandan et al [1] mathematically modelled the steady state heat conduction problem and solved the PDE using FDM method. They implemented the obtained solution using MPI, OpenMP and CUDA. We extend their work by implementing the same solution using hybrid MPI-OpenMP.

## III. METHEDOLOGY

To solve steady state heat conduction problem, we have taken a cube as reference. We assume, material of the cube is homogeneous and there is no heat generating source. Our aim is to find temperature at every point inside the cube at each moment in time. Initially, left face of the cube is at 80° and right face of the cube is at 20°.

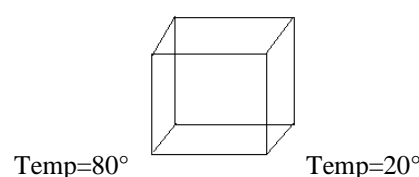


Figure 1: Example Cube

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

Laplacian equation governing the steady state heat conduction problem is given by,

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = 0 \quad (1)$$

The equation (1) is solved using FDM method to obtain numerical solution. FDM method is chosen to calculate temperature at each and every discrete point of the cube at every moment in time. For the cube, number of points in all three directions is assumed to be the same.

Each point inside the cube is represented with its neighbour points is shown in Figure 2

From Figure 2,

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{m+1,n,p} - T_{m,n,p}}{\Delta x} - \frac{T_{m,n,p} - T_{m-1,n,p}}{\Delta x} \quad (2)$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{T_{m,n+1,p} - T_{m,n,p}}{\Delta y} - \frac{T_{m,n,p} - T_{m,n-1,p}}{\Delta y} \quad (3)$$

$$\frac{\partial^2 T}{\partial z^2} = \frac{T_{m,n,p+1} - T_{m,n,p}}{\Delta z} - \frac{T_{m,n,p} - T_{m,n,p-1}}{\Delta z} \quad (4)$$

By (2), (3), (4)

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = \frac{T_{m+1,n,p} - 2T_{m,n,p} + T_{m-1,n,p}}{\Delta x^2} + \frac{T_{m,n+1,p} - 2T_{m,n,p} + T_{m,n-1,p}}{\Delta y^2} + \frac{T_{m,n,p+1} - 2T_{m,n,p} + T_{m,n,p-1}}{\Delta z^2} \quad (5)$$

For steady state heat conduction,

$$\nabla^2 T = 0 \quad (6)$$

$$\Delta x = \Delta y = \Delta z \quad (7)$$

Solving (5) gives,

$$T_{m,n,p} = \left(\frac{1}{6}\right) (T_{m+1,n,p} + T_{m-1,n,p} + T_{m,n+1,p} + T_{m,n-1,p} + T_{m,n,p+1} + T_{m,n,p-1}) \quad (8)$$

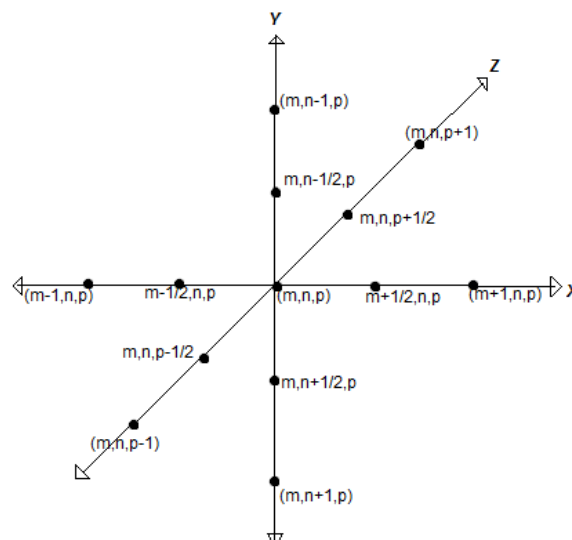


Figure 2: Representation of Single Point along with its Neighbour Points

Hence from above equation, it is shown that, temperature at each point is the average temperature of its neighbour points. Since there are thousands of points inside the cube, computing temperature at each moment in time requires high computational speed. Hence we need to look for a better approach which gives good performance and which is cost effective.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

## IV. OPENMP, MPI AND HYBRID MPI-OPENMP

### OpenMP

It is an application programming interface that enables shared memory parallelism on multi-platform by using its components namely compiler directives, runtime library routines and environment variables. OpenMP is designed to utilize the shared address space most effectively. OpenMP suits well for those environments where all processors effectively utilize the common memory with low communication overhead in the network. For huge-size problems, large shared memory systems are needed, but it is very expensive to manufacture systems with large address space.

### MPI

MPI is the specification of the libraries for message passing. It was designed to accomplish high performance on massively parallel systems. In distributed memory systems, exchange of information between the address spaces of one process to that of another process is made possible using MPI. It uses standard library functions to send and receive messages. Most of the heat conduction algorithms prefer MPI because MPI suits well for large sized problems, but for smaller grids it is not recommendable because of the communication overhead.

### Hybrid MPI-OpenMP

This approach takes the best from both MPI and OpenMP. Major disadvantage of MPI is that communication overhead, but in OpenMP communication among the threads is implicit. Hence latency and communication overhead is reduced in hybrid approach. Hybrid OpenMP-MPI programming model on SMP clusters enables communication within the nodes using OpenMP communication between the nodes using MPI. Hence this is the highly parallelizable, cost effective and most preferable approach for large sized problems.

## V. RESULTS

A 3 dimensional code for FDM solution of steady state heat conduction is developed and then parallelized using OpenMP MPI and hybrid OpenMP-MPI. Table 1 shows the time taken by 3 approaches in seconds for different grid sizes. SMP cluster of 4 nodes has been implemented to run MPI code. Each node is made of Intel® Core™ i5 Processor (Quad core). OpenMP is implemented using 4 threads. From the result, it is observed that, hybrid approach is well suitable for grids with larger sizes.

**Table1: Comparison of Performances of OpenMP, MPI and Hybrid MPI-OpenMP for various Grid Sizes**

Grid Size	Time in seconds				
	OpenMP	MPI(2 nodes)	MPI(4 nodes)	Hybrid OpenMP-MPI(2-nodes)	Hybrid OpenMP-MPI(4-nodes)
25*25*25	0.0447	0.3201	0.3822	0.0308	0.3064
50*50*50	0.7455	2.6672	2.9122	4.0102	1.9244
75*75*75	4.6795	16.5257	16.8757	12.8435	11.0522

## VI. CONCLUSIONS AND FUTURE WORK

A 3-D code for solving steady state heat conduction problem is developed and it has been parallelized using OpenMP, MPI and hybrid OpenMP-MPI. By observing the result, it can be concluded that for larger grid sizes, hybrid approach is better, and for smaller grid sizes, OpenMP suits well. This may be because of efficient usage of shared address space and caches. We extend our work by implementing it using CUDA language on GPU.

## REFERENCES

- [1] Vinaya Sivanandan, Vikas Kumar, Srisai Meher "Designing a parallel algorithm for Heat Conduction using MPI, OpenMP and CUDA" National Conference on Parallel Computing Technologies (PARCOMPTECH), pp-1-7, 2005
- [2] Sonia Rani, Vudutala China V. Rao, Samrit Kumar Maity, Krishan Gopal Gupta "Parallelization of FDM/FEM Computation for PDEs on PARAM YUVA-II Cluster of Xeon Phi coprocessors" Annual IEEE India Conference (INDICON), pp.1-5, 2014

## International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 5, Special Issue 9, May 2016**

- [3] Liu Yan, Zhang Xiong, Lu Mingwan "Meshless Least-Squares Method for Solving the Steady-State Heat Conduction Equation" Tsinghua Science and Technology ISSN 1007-0214 09/18 Volume 10, Number 1 , pp.61-66, 2005
- [4] Nikolaos Drosinos, Nectarios Koziris "Performance Comparison of Pure MPI vs Hybrid MPI-OpenMP Parallelization Models on SMP Clusters". 18th International conference on Parallel and Distributed Processing Symposium ,pp.1-10, 2004.
- [5] Matthew D. Jones, Rutao Yao "Parallel Programming for OSEM Reconstruction with MPI, OpenMP, and Hybrid MPI-OpenMP" Nuclear Science Symposium Conference Record, Volume: 5 IEEE, pp.3036 – 3042, 2004.
- [6] Javier Diaz, Camelia Munˆoz-Caro, and Alfonso Nin "A Survey of Parallel Programming Models and Tools in the Multi and Many-Core Era", IEEE transactions on parallel and distributed systems, vol. 23, No. 8, pp.1369-1386, 2012
- [7] Shikha Mehrotra, Shamjith KV, Prachi Pandey, Asvija B, R Sridharan "A mechanism to improve the performance of Hybrid MPI-OpenMP applications in Grid" High Performance Extreme Computing Conference (HPEC),pp.1-8, 2013
- [8] Miwako Tsuji, Mitsuhsa Sato "Performance evaluation of OpenMP and MPI hybrid programs on a large scale multi-core multi-socket cluster, T2K Open Supercomputer" International Conference on Parallel Processing Workshops, pp.206-213, 2009
- [9] Tutorial on MPI: The Message Passing Interface, <https://www.idi.ntnu.no/~elster/tdt4200-f09/gropp-mpi-tutorial.pdf>