2009 International Conference on Machine Learning and Computing
IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

Mathematical Simulation for 3-Dimensional Temperature Visualization on Open Source-based Grid Computing Platform

Norma Alias [1], Noriza Satam [2], Zarith Safiza Abd. Ghaffar [2], Roziha Darwis[2],

Norhafiza Hamzah [2], Md. Rajibul Islam [1]

[1] Institute of Ibnu Sina, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

[2] Department of Mathematics, Faculty of Science, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

Abstract. New Iterative Alternating Group Explicit (NAGE) is a powerful parallel numerical algorithm for multidimensional temperature prediction. The discretization is based on finite difference method of partial differential equation (PDE) with parabolic type. This paper proposed the NAGE method as a straight forward transformation from sequential to parallel algorithm using domain decomposition and splitting strategies. The processes involving the scheduling of communication, algometric and mapping the sub domain into a number of processors. The critical 3-Dimensional temperature visualization involves large scale of computational complexity. This computational challenge inspiring us to utilize the power of advanced high performance computing resources. By the means of higher performance computing, the computation cannot be relying on just one single set of cluster. Therefore, this research takes the advantage of utilizing multiple set of clusters from geographically different location which is known as grid computing. In realizing this concept, we consider the advantages of data passing between two web services which each are connected with one or multiple set of clusters. For this kind of relationship, we choose service-oriented architecture (SOA) style. Each web services are easily maintainable since there is loose coupling between interacting nodes. The development of this architecture is based on several programming language as it involves algorithm implementation on C, parallelization using Parallel Virtual Machine (PVM) and Java for web services development. As the conclusions, this leading grid-based application platform has a bright potential in managing highly scalable and reliable temperature prediction visualization. The efficiency of this application will be measured based on the results of numerical analysis and parallel performance.

Keywords: parabolic equation, New Alternating Group Explicit (NAGE), parallel algorithm, open source, web services, grid computing

1. Introduction

As a powerful parallel numerical algorithm for multidimensional temperature prediction, New Iterative Alternating Group Explicit (NAGE) will be discretized based on finite difference method of partial differential equation (PDE) with parabolic type.

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} + \mathrm{h}(x, y, z, t), \tag{1}$$

With initial condition,

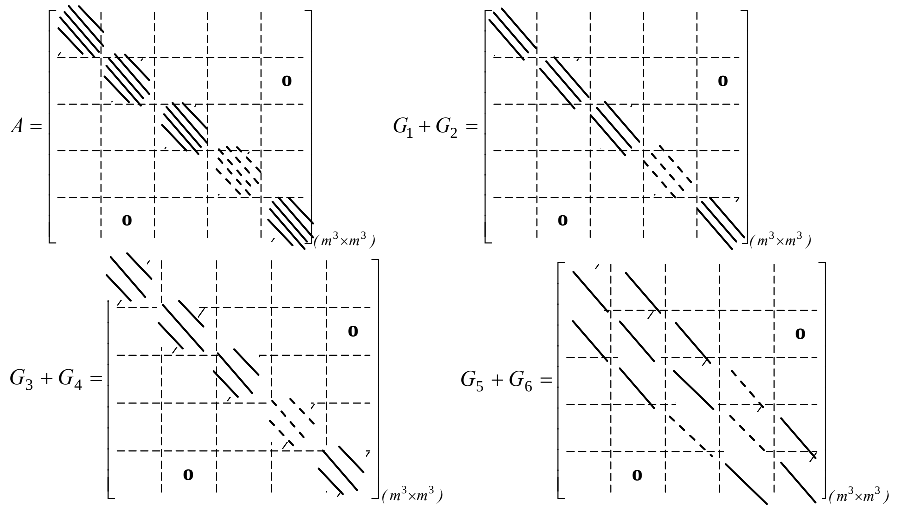
$$U(x, y, z, 0) = F(x, y, z)\ , \quad (x, y, z, t) \in \Re \times 0\,,$$

and boundary condition,

$$U(0, y, z, t) = U(1, y, z, t) = U(x, 0, z, t) = U(x, 1, z, t) = U(x, y, 0, t) = U(x, y, 1, t) = 0.$$

Decomposition of matrix A which is positive and symmetrically defined by

$$\mathbf{A} = \mathbf{G}_1 + \mathbf{G}_2 + \mathbf{G}_3 + \mathbf{G}_4 + \mathbf{G}_5 + \mathbf{G}_6$$



$$(G_1 + rI)u_{[xy]}^{(p+\frac{1}{7})} = (rI - G_2 - G_3 - G_4 - G_5 - G_6)u_{[xy]}^{(p)} + f,$$

$$(G_2 + rI)u_{[xy]}^{(p+\frac{2}{7})} = G_2 u_{[xy]}^{(p)} + r u_{[xy]}^{(p+\frac{1}{7})},$$

$$(G_3 + rI)u_{[xy]}^{(p+\frac{3}{7})} = G_3 u_{[xy]}^{(p)} + r u_{[xy]}^{(p+\frac{2}{7})},$$

$$(G_4 + rI)u_{[xy]}^{(p+\frac{4}{7})} = G_4 u_{[xy]}^{(p)} + r u_{[xy]}^{(p+\frac{3}{7})},$$

$$(G_5 + rI)u_{[xy]}^{(p+\frac{5}{7})} = G_5 u_{[xy]}^{(p)} + r u_{[xy]}^{(p+\frac{4}{7})},$$

$$(G_6 + rI)u_{[xy]}^{(p+\frac{6}{7})} = G_6 u_{[xy]}^{(p)} + r u_{[xy]}^{(p+\frac{5}{7})},$$

$$u_{[xy]}^{(p)} = u_{[xy]}^{(p)} + 2\left(u_{[xy]}^{(p+\frac{6}{7})} - u_{[xy]}^{(p)}\right). \quad (2)$$

Iteration of this method will stop until the slave had met the local stopping criterion that is, $\left|u_{i,j,k}^{(p+1)}(p)\right| \le \varepsilon(p)$ and master met the global converge criterion, ε.

2. Grid Computing Platform

As an advancement of conventional supercomputer, the real and specific problem that underlies the grid concept is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations [5]. This research proposed grid-based software that aim to handle problems which have high rate of computational complexity by providing large space of memory allocation, minimum consumption of multiprocessor system cost as well as maintainable cluster of workstations that is distance located.

The 3-Dimensional computational challenge drives the inspiration to utilizing the high ability rate of computing resources. By the means of higher performance computing, the computation cannot be relying on just one single set of cluster. Therefore, this research takes the advantage of utilizing multiple set of clusters from geographically different location which is known as grid computing. In realizing this concept, we consider the advantages of data passing between two web services which each are connected with one or multiple set of clusters. Each web services are easily maintainable since there is loose coupling between interacting nodes. For this kind of relationship, we choose Service Oriented Architecture (SOA) framework. As, grid computing is aimed at sharing dynamically heterogeneous resources, SOA plays role as a style that emphasize business ability to be run in an interoperable way [2].

The use of SOA ensure continuous system improvement and able to group various functionality as interoperable services. This framework allows embedded web services to exchange data between one another during parallel processing execution. The ideas of SOA developed based on previous perceptions of modular programming and distributed computing [4].

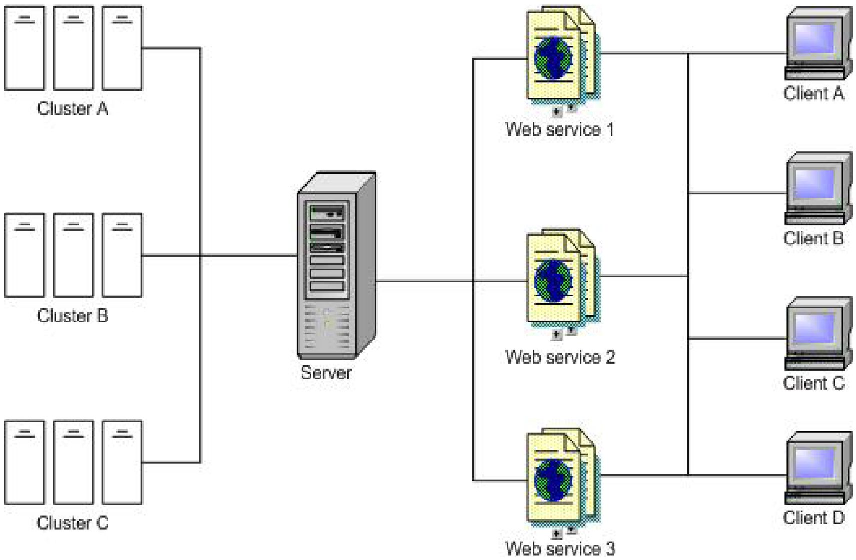


Figure 1: Architecture of 3-dimensional grid-based temperature visualization solver

Figure 1 shows the architecture of 3-dimensional grid-based temperature visualization solver. The development of this architecture is based on numbers of programming language as it involves algorithm implementation on C, parallelization using Parallel Virtual Machine (PVM) and Java for web services development. In this research, Web Services Definition Language (WSDL) is being selected as web services XML standardized technology and developed using NetBean IDE tools. Glassfish 2.0 will be use as application server for this architecture. The grid computing platform is an open source-based and will be develop under Linux environment. This platform development will accelerate the speeds of execution and scaled-out across a virtualized grid. The clusters of processors involved in this platform are developed on increasingly larger computational hardware with inexpensive specifications.

3. Numerical Analysis

Numerical analysis of the NAGE covers the area of numerical result, communication cost, computational complexity and 3-dimensional temperature visualization. A numerical result based on sequential algorithm in solving the equation (1) is given in Table 1. Red-Black Gauss Seidel (RBGS) is used to make comparison between classical iteration method and new iterative method.

Table 1: Numerical result for sequential NAGE and RBGS method

$m \times m \times m$	$100 \times 100 \times 100$		$140 \times 140 \times 140$	
Method	*NAGE*	*RBGS*	*NAGE*	*RBGS*
execution time (μs)	216.3390	282.9480	650.7960	558.9310
no. of iteration	115	760	110	650
mean square error	5.9814 E^{-7}	4.1178 E^{-5}	8.0438 E^{-8}	8.0842 E^{-8}
absolute error	9.8683E^{-10}	1.3878 E^{-17}	8.3312 E^{-11}	5.5511E^{-17}
root mean square error	6.5119 E^{-13}	4.4789 E^{-5}	1.2012 E^{-14}	1.2012 E^{-14}
maximum error	2.2480 E^{-6}	2.2521 E^{-6}	3.0667 E^{-7}	1.0152 E^{-11}
Δx	1.0000 E^{-2}	1.0000 E^{-2}	7.1428 E^{-3}	7.1428 E^{-3}
Δy	1.0000 E^{-2}	1.0000 E^{-2}	7.1428 E^{-3}	7.1428 E^{-3}
Δz	1.0000 E^{-2}	1.0000 E^{-2}	7.1428 E^{-3}	7.1428 E^{-3}
Δt	6.6667 E^{-5}	6.6667 E^{-5}	2.000 E^{-5}	2.000 E^{-5}
convergence criterion	1.0 E^{-9}	1.0 E^{-9}	1.0 E^{-9}	1.0 E^{-9}

Table 2 shows the communication cost of parallel NAGE for different matrix size (m) .

From Table 1, it can be observed that NAGE converged faster than RBGS and this situation led to less time consumed in obtaining final value.

Table 2: Communication cost for the parallel NAGE and parallel RBGS method

$m \times m \times m$	$100 \times 100 \times 100$	$140 \times 140 \times 140$
Method	*Communication cost*	*Communication cost*
NAGE	$1380(m \times m)t_{data} + 690(t_{start} + t_{idle})$	$1320(m \times m)t_{data} + 660(t_{start} + t_{idle})$
RBGS	$9120((m \times m)/2)t_{data}$ $+4560(t_{start}+t_{idle})$	$7800((m \times m)/2)t_{data}$ $+3900(t_{start}+t_{idle})$

Table 2 shows the communication cost for NAGE and RBGS for different matrix size. Conclusion can be made that parallel NAGE method able to solve larger problem with low rate of communication cost. For larger matrix size, NAGE performs well where it reduces the communication cost during program execution.

In measuring computational complexity of NAGE, the overall needs of basic computation operations are counted. The analysis is as summarized in Table 3 where NAGE produces less computational complexity compared to RBGS.

Table 3: Computational complexity for the parallel NAGE and parallel RBGS method

$m \times m \times m$	$100 \times 100 \times 100$		$140 \times 140 \times 140$	
Method	*Addition*	*Multiplication*	*Addition*	*Multiplication*
NAGE	$[2875(m\text{-}1)^3$ $+2875m^3+1725m^2]/p$	$[1380(m\text{-}1)^3$ $+3450m^3+1610m^2]/p$	$[2750(m\text{-}1)^3$ $+2750m^3+1650m^2]/p$	$[1320(m\text{-}1)^3$ $+3300m^3+1540m^2]/p$
RBGS	$[9880(m\text{-}1)^3$ $+9120m^3]/p$	$[11400(m\text{-}1)^3$ $+10640m^3]/p$	$[8450(m\text{-}1)^3$ $+7800m^3]/p$	$[9750(m\text{-}1)^3$ $+9100m^3]/p$

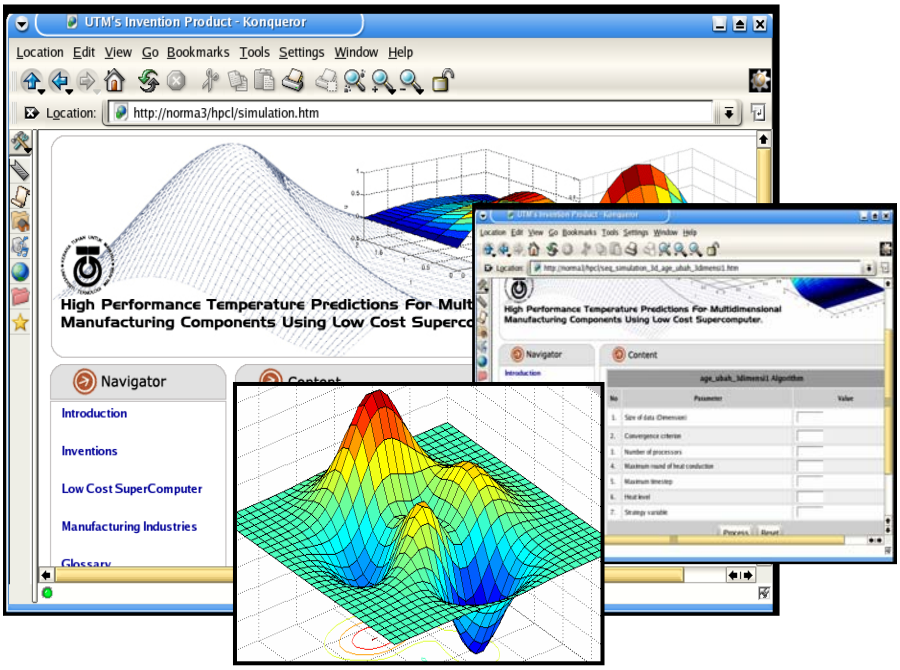
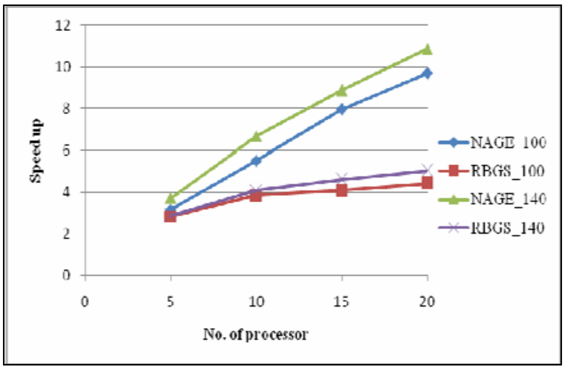


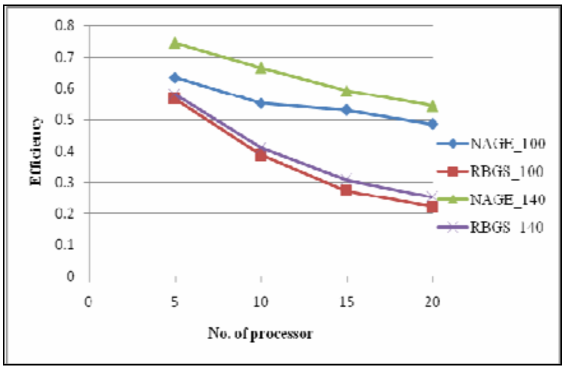
Figure 2: Simulation results and snapshot of web based application

4. Performance Measurement of Visualization Process

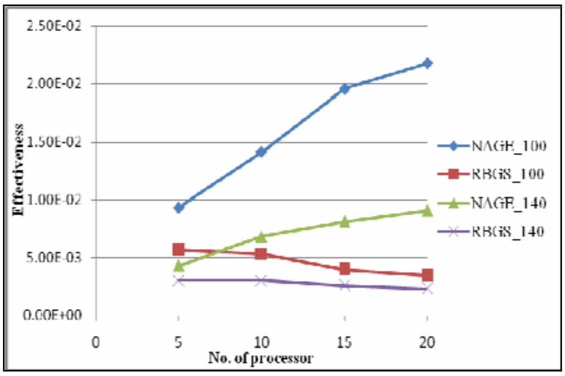
Grid computing technology gives significance positive impact to the performance of visualization performance. The visualization process successfully accelerate to become extremely fast while maintaining its reliability and accuracy of the outcomes. The performance of parallel algorithm and application can be measured in terms of speedup, efficiency, effectiveness and temporal performance which used the following definitions, speed up, $S_p = \frac{T_1}{T_p}$, efficiency, $C_p = \frac{S_p}{p}$, effectiveness, $F_p = \frac{S_p}{C_p}$ and temporal performance $L_p = \frac{1}{T_p}$ where T_1 and T_p are execution time for parallel algorithm. Figure 4 shows the performance measurement analysis graph plotted against number of processors. Figure 6(a) shows the speedup increment upon addition of more processors to the system. Performance increment can also be observed when measuring effectiveness and temporal performance. This reflects that adding more processors will potentially increase the performance of parallel processing system. On contrary, increment of processors numbers decreases the efficiency rate. According to [3] increasing of processors reduces the total execution time, and subsequently decreases the rate of efficiency. The reduction of efficiency rate is due to situation where each processor responsible for fewer tasks as processors numbers increase with constant problem's size. Thus, efficiency will increase for increasing problem size with constant number of processors.



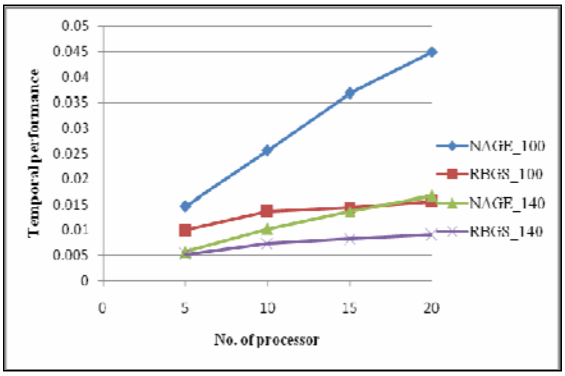
3(a) Speed up



3(b) Efficiency



3(c) Effectiveness



3(d) Temporal performance

Figure 3: Parallel performance evaluation for temperature visualization

NAGE method performs better than RBGS especially in larger matrix size. It can be conclude that this new method able to solve larger problem while maintaining its performance. The evaluation of parallel performance shows that NAGE method has bright potential compared to classical iterative method namely RBGS and brings the performance to a better stage especially in solving large size of problem.

5. Conclusion

This research had experimentally proved that the grid-based NAGE method has significance performance improvement over the locally based parallel execution. This 3-dimensional temperature visualization solver available to produce highly accurate results and can be use to solve any size of problems. As currently the clusters only using various kinds of processors ranging from Pentium III to dual-core, we are looking forward for future advancement of the cluster's power by improving the Random Access Memory (RAM) capability and the memory storage of each processor. Continuous system upgrades will enhance the performance of the heterogeneous clusters to a better stage. Grid computing advantages flavored the grid-based application reliability as other clusters will cover the performance defects in case of cluster failure. This is merely different with traditional distributed system where each processing element (PE) is considered as always available and burden the developer to ensure the PE to be in ready-to-use condition.

6. Acknowledgement

The authors acknowledge Institute of Ibnu Sina, UTM and Ministry of Science, Technology and Innovation, Malaysia (MOSTI) for the financial support under research vote 79219.

7. References

[1] H. Nathan, W. Burkhard, T. Ewan, “A Framework for Interactive Web-Based Visualization”, *Seventh Australasian User Interface Conference (AUIC2006)*, *Conferences in Research and Practice in Information Technology (CRPIT),* Vol. 50. Wayne Piekarski, Ed, 2006.

[2] L. Domenico, Via G. Moruzzi, Fabrizio Silvestri, *Ranking of Grid-enabled Workflow Service Providers*, HPDC’08, June 23–27, 2008, Boston, Massachusetts, USA. ACM 978-1-59593-997-5/08/06.

[3] A.Y. H. Zomaya, Parallel and Distribution Computing Handbook (MacGraw Hill, 1996).

[4] Norma Alias, Suhaimi Mydin, Md. Rajibul Islam, Norhafiza Hamzah, Zarith Safiza Abd. Ghaffar, Noriza Satam, Roziha Darwis, ”Grid Portal Technology for Web based Education of Parallel Computing Courses, Applications and Researches”. in Proc. of *The 8th International Conference on Web-based Education (WBE 2009)*, Phuket, Thailand, Mar. 16-18, 2009, pp. 319-324.

[5] Bing Wu, Matthew Dovey, Muan Hong Ng, Kaihsu Tai, Stuart Murdock, Hans Fangohr, Steven Johnston, Paul Jeffreys, Simon Cox, Jonathan W. Essex and Mark S.P. Sansom, “A Web / Grid Portal Implementation of BioSimGrid: A Biomolecular Simulation Database”, *Journal of Digital Information Management, Vol. 2* (2), 2004, 74-78.

[6] Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., Sunderam, V., *PVM: Parallel Virtual Machine & User's Guide and Tutorial for Networked Parallel Computing*, (MIT Press, Cambridge, Mass, 1994).

[7] A. D. Polyanin, Handbook of Linear Partial Differential Equations for Engineers and Scientists, Chapman & Hall/CRC Press, Boca Raton, 2002.