www.arpnjournals.com

HIGH-PERFORMANCE COMPUTING AND COMMUNICATION MODELS FOR SOLVING THE COMPLEX INTERDISCIPLINARY PROBLEMS ON DPCS

Norma Alias[1], Riadh Sahnoun[1] and Victor Malyshkin[2]

[1]Ibnu Sina Institute for Fundamental Science Studies, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia
[2]Supercomputer Software Department, Institute of Computational Mathematics and Mathematical Geophysics Siberian Branch of Russian Academy of Sciences, Russia
E-Mail: normaalias@utm.my

ABSTRACT

The paper presents some advanced high performance (HPC) and parallel computing (PC) methodologies for solving a large space complex problem involving the integrated difference research areas. About eight interdisciplinary problems will be accurately solved on multiple computers communicating over the local area network. The mathematical modeling and a large sparse simulation of the interdisciplinary effort involve the area of science, engineering, biomedical, nanotechnology, software engineering, agriculture, image processing and urban planning. The specific methodologies of PC software under consideration include PVM, MPI, LUNA, MDC, OpenMP, CUDA and LINDA integrated with COMSOL and C++/C. There are different communication models of parallel programming, thus some definitions of parallel processing, distributed processing and memory types are explained for understanding the main contribution of this paper. The matching between the methodology of PC and the large sparse application depends on the domain of solution, the dimension of the targeted area, computational and communication pattern, the architecture of distributed parallel computing systems (DPCS), the structure of computational complexity and communication cost. The originality of this paper lies in obtaining the complex numerical model dealing with a large scale partial differential equation (PDE), discretization of finite difference (FDM) or finite element (FEM) methods, numerical simulation, high-performance simulation and performance measurement. The simulation of PDE will perform by sequential and parallel algorithms to visualize the complex model in high-resolution quality. In the context of a mathematical model, various independent and dependent parameters present the complex and real phenomena of the interdisciplinary application. As a model executes, these parameters can be manipulated and changed. As an impact, some chemical or mechanical properties can be predicted based on the observation of parameter changes. The methodologies of parallel programs build on the client-server model, slave-master model and fragmented model. HPC of the communication model for solving the interdisciplinary problems above will be analysed using a flow of the algorithm, numerical analysis and the comparison of parallel performance evaluations. In conclusion, the integration of HPC, communication model, PC software, performance and numerical analysis happens to be an important approach to fulfil the matching requirement and optimize the solution of complex interdisciplinary problems.

Keywords: high-performance computing, communication models, interdisciplinary problems, mathematical modelling and numerical simulation.

INTRODUCTION

Nissani [1], highlighted ten reasons on why the integrated, interdisciplinary knowledge and research were classified as priority needs. Interdisciplinary appears as the complex systems, involving a global behavior with multi-parameters that belong to any discipline and interacting via local rules. Based on these problems, the hierarchies for solving the complex behavior can be organized in a dynamical process reconstructing the complex system from a big data set and controlling the evolution. The main objective of solving the complex interdisciplinary problem is to predict, monitor, and observe, visual and robust systems to perturbation of the future behavior and to design dedicated systems having the same kind of mechanical properties [2]. Thus, the complex interdisciplinary problem can be solved by the combination of different models such as an inverse problem, cellular automata, dynamical systems, partial differential equations (PDE), the hybrid model and artificial neural network (ANN) [3-4]. The domain of solution is dealing with large sparse linear systems and required a wide-range power simulation tools, huge memory allocation, high speed processors, fast computational process and limited communication cost.

Complex interdisciplinary solver on HPC

This paper explains the HPC and communication model to improve the scientific computing performance by utilizing the combination of PC, DPCS, and memory storage and network technology to meet the objective mentioned above. In nanotechnology, various physical processes which carry out sensitivity experiments involve a large-scale structure of biomolecule modeling and reformation of the nanoscale system [5]. For example, Figure-1 shows the visualization of the nanoparticle for drug delivery system using biomolecular and reformation modeling in COMSOL Multiphysic software. A power system of HPC solutions is required to support the nanoscale computation and to produce a highly scalable system on PC architecture. MP4 calculations, orbital shapes, HOMO-LUMO energy gaps and spectral properties of the molecule presents the prediction of

www.arpnjournals.com

physical and chemical properties of small to medium size molecular structures such as benzene [6]. This paper presents the performance of LINDA programming language on Multicores processor, 2X3TB, WD BLACK CAVIAR, 16GB RAM for predicting chemical properties of benzene. In science and engineering, big data are needed for process optimization, process control and product design, as shown in Figure-2. The curing and molding processes for tire tread manufacturing involve six interdisciplinary processes. These comprise of heat curing to accelerate the cure of rubber, modeling the process using improved Stefan problem coupled with the Navier–Stokes equations, applying domain decomposition techniques for data partitioning, configuring the DPCS system, implementing the parallel algorithm on DPCS, visualizing the parameter changes and cooling process prediction to ambient temperature. Based on multidimensional PDE modeling, finer grain data in simulation has been extensively applied on HPC in order to increase the speedup of computation of a big data simulation. In systems integration and software design, the 3D animation, visualization and movement in large basins recognized as an important outcome [7]. PC systems deliver the required tools to achieve the real time solution. In image processing, HPC accurately predicts on high-resolution images and visualizing the physical oceanography in huge scale as shown in Figure-3. In agriculture, large scale environmental modeling to predict and to observe the future environment considered as a real time solution. Thus, indicates the capability of HPC to increase the speedup of the nanoscale computational process. In urban planning, HPC serve as a required approach in simulating the land cover dynamics, intersecting the emissions processing and analyzing the spatial, temporal segmentation within a huge number of grid cells of Geographic Information Systems (GIS). Figure-4 shows the spatial, temporal segmentation and visualization to produce a layered image of the urban area, Nusajaya, Malaysia based on finer grid cells of GIS [8]. Figure-5 shows the high-performance edge detection model for an early stage of brain tumor growth. Figure-6 shows the ability of the CPU-GPU platform with CUDA parallel programming to simulate a big data of biomedical dynamic modeling integrated with ANN for eye movement and eye muscle stress problem. There involve seven integrated steps to develop HPC software based on a parallel algorithm and the numerical library of complex problems, as shown in Figure-7. This Figure-7 shows an example of an embedded system of printed circuit board derived from the development of HPC software. There are several advanced knowledge requirements in developing prototypes of HPC software. These requirements include data collection from the printed circuit board, converting signals to digital data set, presenting domain of solution in 2D, decomposing the domain into smaller non-overlapping subdomains, designing the communication activities among processors, implementing parallel algorithms on DPCS, developing the HPC software using open source software such as Perl-CGI, HTML, PHP and MySQL database as the web development tools. HPC

serves as a key enabling technology of the scientific computing with high-performance evaluations. The communication model provides a platform for exchanging data among the processors of the HPC system.
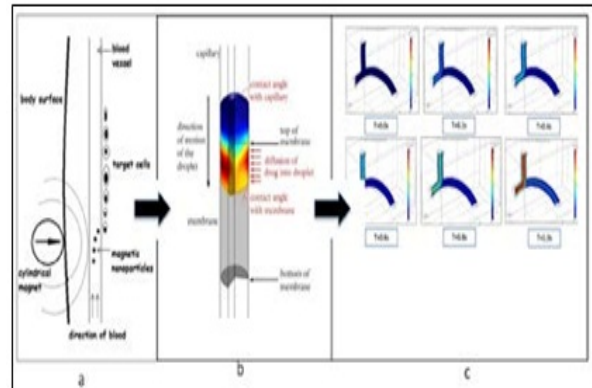

Figure-1. The visualization of nanoparticle for drug delivery system using biomolecular and reformation modelling in the COMSOL Multiphysic software.
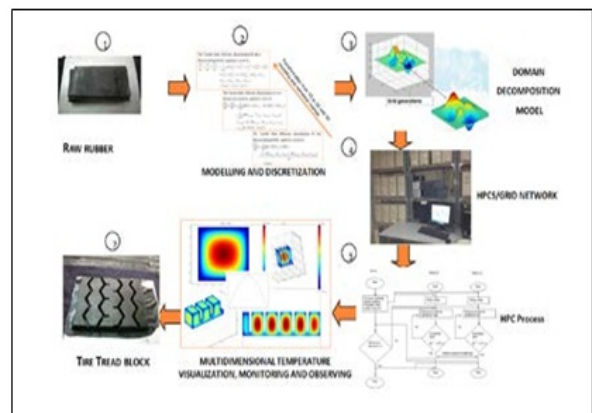

Figure-2. The process optimization, process control and product design for tire tread molding.
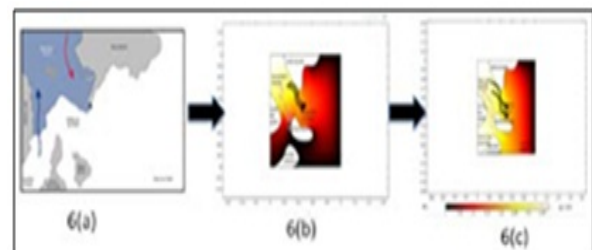

Figure-3. Visualization of the physical oceanography in huge scale of wind direction, ocean movement and temperature behavior of Ross ice shelf, Antarctic region.
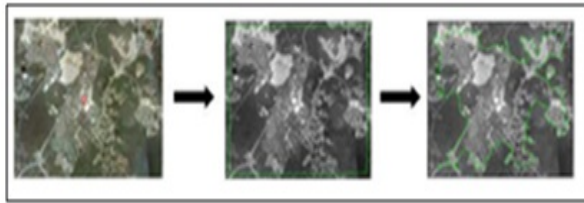
Figure-4. Spatial, temporal segmentation and visualization to produce a layered image of the Nusajaya urban area based on finer grid cells of the Geographic information systems (GIS).
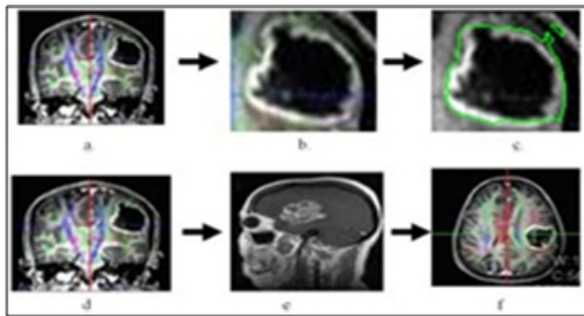
Figure-5. Reconstruction of MRI images for high resolution of brain tumor growth.
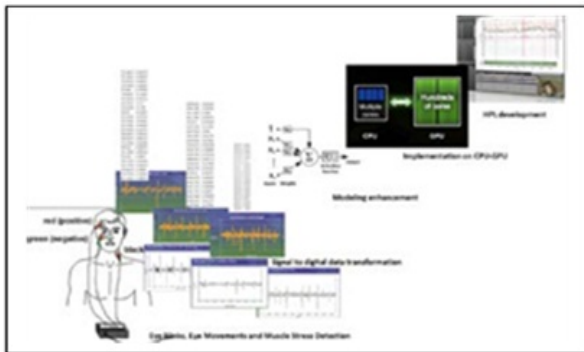
Figure-6. Parallel biomedical dynamic modeling integrated with ANN for eye movement and eye muscle stress problem using CPU-GPU platform.
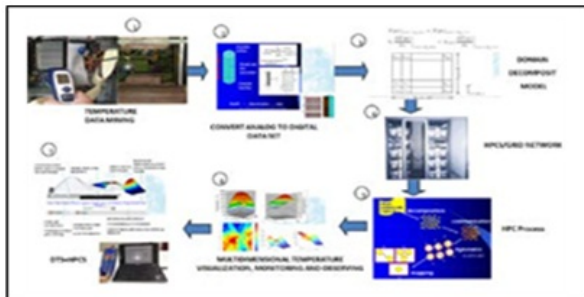
Figure-7. The HPC Software engineering development contains the numerical library and repository on DPCS platform.

Communication models of HPC systems

The communication models involve the synchronization of the data and communication networks between the different subdomains. The communication models of parallel programming languages can be categorized as a client-server model, slave-master model, server-worker model and fragmented model to perform the parallelism of the interdisciplinary problem [9-10]. Some parallel programming languages under consideration are PVM, MPI, LUNA, MDC, OpenMP, CUDA and LINDA.

Communication model of PVM is based on master-slave model and enables interprocessor communications. Task-to-task communication is done with the message passing paradigm and PVM is considered as a high-level interactive language. Thus, the computing platforms will be able to deliver high levels of performance and functionality. This model is well suited for solving the interdisciplinary problems using a huge number of iterations that converge to the exact solution, but with high computational cost and involves high communication activities. Communication model of MPI is based on the message-passing model between client and server processes. In the message-passing model, the tasks are separate processes that communicate synchronously and explicitly via the high speed network. Communication model of MDC is based on server-worker model. The interactive parallel computation is referring to how speed and how efficient the running workers complete the computation on DPCS and the ability to connect the user desktops. MDC uses MPI as an engine to support the communication activities. Communication model of OpenMP transpires to be a classical process-to-process communication model in hybrid solutions. The platform depends on the existence of multiple threads or Multicores in the shared memory architecture. Thus confirm the program of OpenMP as a shared memory programming model. Communication model of LINDA software founded on a model of coordination and communication among processes using domain decomposition strategy on shared and virtual memory. This model functions as a coordination language, where several operating primitives are added to a sequential language to support a message-passing paradigm. Fine grain in data simulation and high communication cost fits well on LINDA. The superiority of LINDA comes from the fine grain, which being implemented in shared memory architecture, thus benefiting most molecular modeling software. On the other hand, LUNA software is classified as high performance computation in the programmable fragmented technology. In LUNA, the development of fragmented program, compilation and execution are used to optimize the execution. The fragmented programming language and the run-time system include methods and algorithms of optimizations [11]. Communication model of the CUDA software depends on the block and thread structure on shared memory architectures. For example, two threads from different blocks can communicate with each other. The combination of multi-core CPU and graphics processing units (GPU) device can communicate

via shared host memory and via host-side CUDA message passing paradigm. PVM, MPI, MDCS, OpenMP, CUDA and LINDA depend on the integrated parallel programs between computation cost and communication activities. LUNA fragmented algorithm depends on the integrated parallel programs between data and computation fragments. The theory of structural synthesis differentiates the data and computation fragments before designing implementing the communication strategy. This paper will emphasize on how the communication model can efficiently be matched to compute the specific mathematical modeling in a complex system.
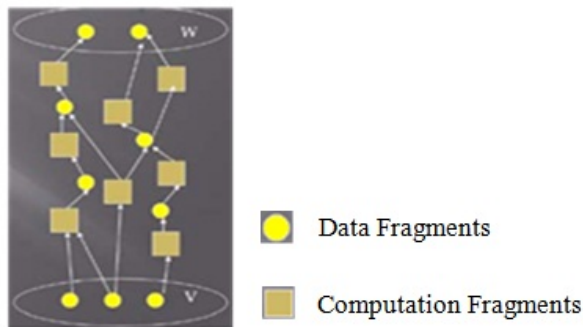
Figure-8. The integrated data and computation fragments.

Mathematical modeling

The high-quality interdisciplinary experiment needs the collaboration, combination of sub-work experiments, high cost and time consuming. In the case of limited resolution and resources, the outcome would definitely be unpredictable.

Integrated mathematical modeling and simulation strategies have the potential to overcome those limitations. Predictive models with experimental observations, inverse problem, cellular automata, dynamical systems, PDE, hybrid and ANN modeling capabilities to solve the complex interdisciplinary problem. The governing equations for biomolecular modeling of the nanoparticle for drug delivery system via blood flow combined the mass continuity and momentum conservation equation. The flow considered is the influence of externally applied magnetic field in an axial direction. Assuming the momentum equations in the cylindrical coordinate system helps maintain its continuity (r, z, θ) [12]. The mathematical modeling the optimization process and thermal process shown in Figure-2 builds on PDEs with parabolic type. Using the improved Stefan-model coupled with the Navier –Stokes equations, a change in the dependent variable of the PDEs by the formula leads to the achievement of the optimization of temperature behavior [13]. Physical oceanography in a huge number of grid cells from GIS images and large-scale atmospheric circulation is important to understand the natural phenomena of ocean circulation [14]. Figure-3 illustrates the observation of wind direction from the Ross Ice Shelf, Antarctica region. On the other hand, the large scale of the integrated modeling of GAC, AOS and ANN is the alternative method to govern the spatial, temporal

segmentation and visualization images of the Nusajaya region in Malaysia, as shown in Figure-4 [8]. Furthermore, in an earlier study carried out by our group, we have proposed the reconstruction of MRI images using edge detection model and PDE with elliptic type, as shown in Figure-5. The image manifold and volume estimation methods are needed for the 3D reconstruction of tumor images from a set of 2D MRI images. In this case, HPC serves as the main platform to support the visualization and animation. The stress and strain of the muscle problem can also be mimicked by the elasticity as shown in Figure-6 [15]. It should be noted that large sparse elements of FEM are generated to monitor the displacement and stresses of the muscle. Figure-7 shows the computation engine for HPC software development based on mathematical modeling of multidimensional PDE with elliptic and parabolic types. It demonstrates that the huge memory allocation of DPCS platform supports the repository and numerical library of HPC software [13].

Numerical simulation

Several numerical methods offer a straightforward implementation of converting the mathematical model into a set of grid points. FDM, FEM and direct methods work as tools to discretize the mathematical modeling and to generate large sparse linear systems. Numerical simulations of explicit, implicit and Crank-Nicolson schemes fit well in solving the large sparse matrix of linear systems. Finer scale modeling, decomposition techniques, aggregation and mapping processes are able to obtain the parallel simulation of the large-scale modeling. Thus, a wide-range power simulation tools are required for handling a very large sparse matrices, huge memory allocation and high-speed processors of HPC systems.

SEQUENTIAL & PARALLEL IMPLEMENTATIONS

In this section, we mainly focus on COMSOL and C++/C to support the sequential algorithm for solving a small scale of the numerical simulation. The integrated PVM/MPI/CUDA and C++/C, LUNA, MDC, OpenMP, and LINDA are used to develop the parallel program of the huge scale numerical simulation.

Sequential algorithm

Sequential algorithm is an algorithm that can be executed serially using a single processor. Based on the properties of time, T and number of processors, N, the sequential algorithm can be defined as,

$$T_{sequential} = T_{parallel} \cdot N_{processor} \qquad (1)$$

The sequential algorithm shown in Figure-9 fit well to be written in c programming languages, particularly c++.
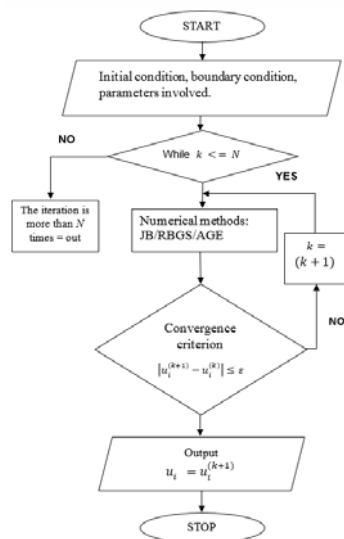
Figure-9. Sequential algorithm of the numerical simulation.

Parallel algorithm

Parallel algorithm specifies as an integrated algorithm that can be executed concurrently at a time using many processors. Parallel algorithm is defined as,

$$T_{parallel} = \frac{T_{sequential}}{N_{processor}}$$

(2)

The parallel algorithm shown in Figure-10 fits well to be implemented in a programming language such as PVM, MPI, CUDA, LUNA, MDC, OpenMP, and LINDA. Parallel algorithm has a high potential to simulate a finer resolution mesh using domain decomposition techniques with feasible and cost effective on HPC. The choice of the specific communication model, the design of communication activities, the parallel performance analysis, the development and implementation of parallel algorithms on DPCS are still challenging areas.
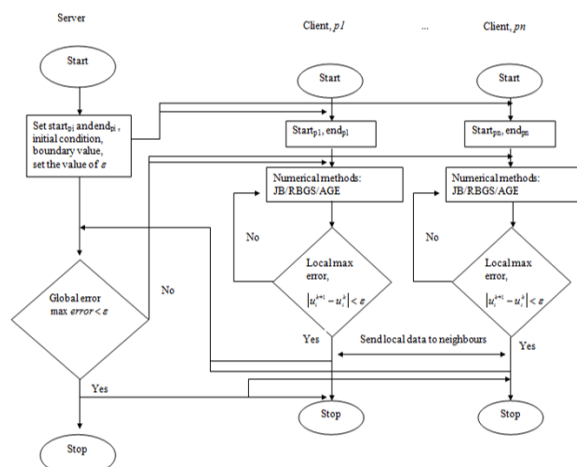

Figure-10. Parallel algorithm for the numerical simulation.

Implementation on DPCS

DPCS is developed based on Beowulf cluster from low cost CPU hardware and supply arguments in a single instruction and exploits multiple data streams (SIMD) style of parallel processing as shown in Figure-11. The processing power of multicore CPU and GPU results in the speedup of scientific execution, as shown in Figure-3. Most of the independent GPU devices come from AMD, Intel and NVIDIA. The combination of CPU and GPU designed for handling multiple tasks simultaneously by using CUDA programming language. The parallel program such as PVM, MPI, LUNA, MDC and OpenMP are generally stable when executed on DPCS.

Figure-11. DPCS based on Beowulf cluster development.

PERFORMANCE EVALUATIONS & DISCUSSIONS

Performance evaluations

There are two categories to measure the performance of complex modeling on the HPC platform. The first is the performance evaluation of PC and the second is the numerical analysis of the mathematical modeling. Parallel performance evaluations include run time, speedup, efficiency, effectiveness and temporal performance. [16] defined the speedup, S_p as

$$S_P = \frac{1}{F + \frac{(1-F)}{P}}$$

(3)

where P is the number of processors, F is the fraction of sequential computation while $(1-F)$ is the fraction of parallel computation. The ratio of speedup to number of processors is defined as efficiency, E_p, where

$$E_P = \frac{S_P}{P}$$ and effectiveness, V_p, where $V_P = \frac{S_P}{E_p}$

Computational complexity is referred to the arithmetic and algebraic functions such as addition, subtraction, multiplication, division, square root and polynomial functions. The critical issue and giving impact to the parallel performance evaluation is the communication cost and its activities. The communication cost is dependent on the programming semantics, the network topology, data handling and the communication model associated with the software protocols. The measurement of numerical analysis for complex mathematical modeling is based on run time, accuracy, convergence, root means square error and the number of

iterations. In this paper, the PDEs model for solving the complex interdisciplinary problems is proven to be highly stable, unique and consistent.

NUMERICAL RESULTS AND DISCUSSION

The magnetic nanoparticles for drug delivery and drug release models are assigned to develop the alternative numerical simulation of PDEs as shown in Figure-1. Some numerical schemes such as Jacobi (JB), Gauss Seidel (GS) and Red, Black Gauss Seidel (GSRB) influenced the FDM. Table-1 shows the superiority of the GS scheme in terms of faster convergence, higher accuracy and lower computational cost for solving the PDEs via FDM strategies.

Table-1. The comparison of numerical analysis of drug delivery and drug release models using JB and GS schemes.

| Numerical Analysis		Drug delivery		Drug release	
		Jacobi	Gauss Seidel	Jacobi	Gauss Seidel
Time Execution(s)		0.001345	0.001072	0.000434	0.000086
Iteration		108	66	34	19
Convergence Rate		1	0.671	1	0.638
computational complexity	multiplication	183040	118592	56320	33440
	additional	160160	80251	49280	29260
Absolute Error		1.70439e-5	10.85295e-6	7.75848e-6	9.54395e-6
RMSE		1.48924e-5	5.56607e-6	2.66464e-6	4.76947e-6

Figure-12 shows some numerical results, 2D and 3D visualization of curing process using phase change simulation of the improved Stefan problem coupled with the Navier–Stokes equations as stated in Figure-2.
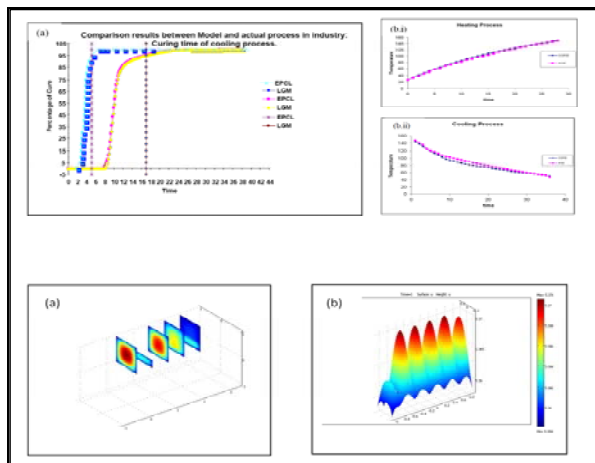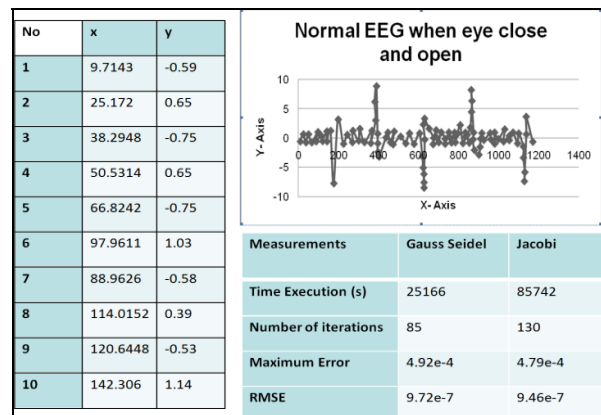

Figure-12. (a) Comparison of curing time between mathematical modeling and industry real data (LGM). (b) Comparison of heating and cooling process between two numerical schemes, GS and AGE. (c) Comparison of two and three dimensional for the temperature distribution in COMSOL Multiphysic.
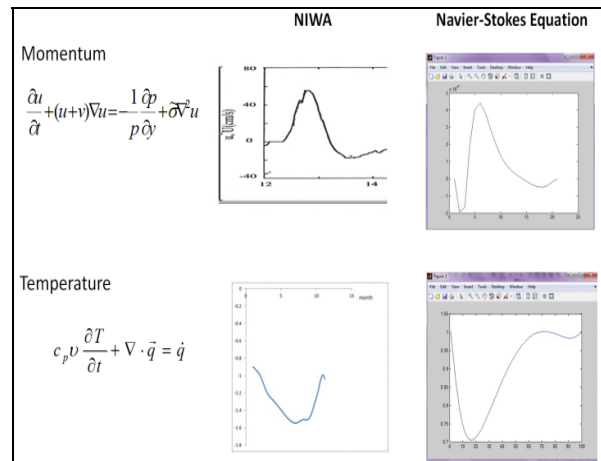
Figure-13. Comparison of graphs between Navier-Stokes equations and industry real data (NIWA) in terms of momentum, temperature behavior and salinity of Ross Ice shelf, Antarctic region as stated in Figure-3.
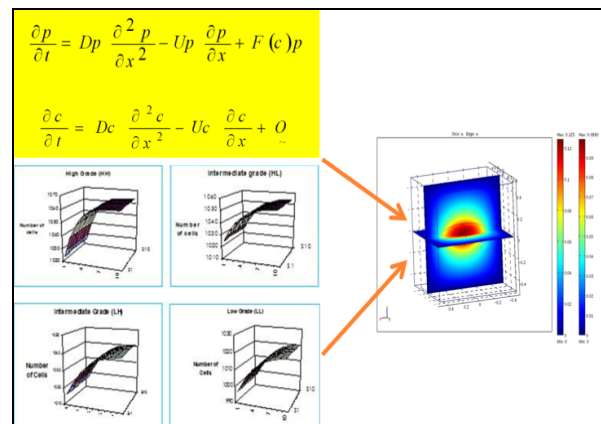

Figure-14. Transformation from 2D brain tumor model to 3D visualization using mathematical modeling for enhancing the problem as shown in Figure-5.

No	x	y		
1	9.7143	-0.59		
2	25.172	0.65		
3	38.2948	-0.75		
4	50.5314	0.65		
5	66.8242	-0.75		
6	97.9611	1.03		
7	88.9626	-0.58	Measurements	Gauss Seidel	Jacobi
8	114.0152	0.39	Time Execution (s)	25166	85742
9	120.6448	-0.53	Number of iterations	85	130
10	142.306	1.14	Maximum Error	4.92e-4	4.79e-4
			RMSE	9.72e-7	9.46e-7

Figure-15. Some numerical results for future prediction of parallel biomedical dynamic modeling integrated with ANN for muscle stress problem and eye movement as shown in Figure-6.

ARPN Journal of Engineering and Applied Sciences

The conceptual diagram of HPC software consists the interaction process among the users, algorithm provider and user interface. Based on software engineering flow as stated in Figure-7, there are three different modules according to their functionality. The modules are 'Authoring, Publishing and Supporting'. Figure-16 shows there are several levels to access the HPC software such as allow the common users browse the code, view tutorials and explore the code. The functions of scientific librarian are to approve the status of a contribution of algorithm provider and publish the contribution code. A huge virtual memory and high-speed processors support the repository and the numerical library of HPC software as shown in Figure-16 [13].
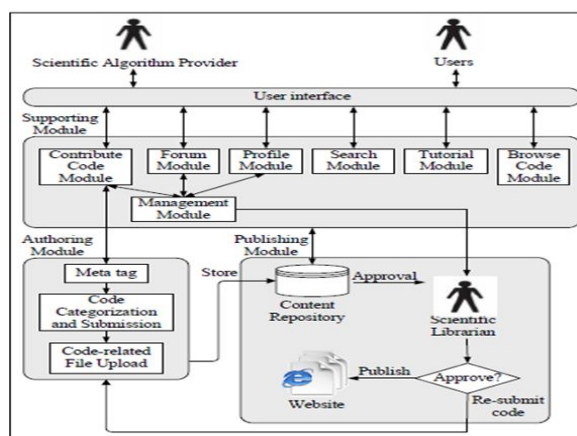
Figure-16. Conceptual model of HPC software development.

PARALLEL PERFORMANCE RESULTS AND DISCUSSIONS

The parallel performance evaluations are referred to the complex interdisciplinary in section 1. In Table-2, the parallel algorithm of drug delivery and drug release models shows that the run times consistently decreases as the number of threads used increase for every size of grid points.

Table-2. Run time of the parallel algorithm for drug delivery and drug release models versus number of threads. The mesh sizes correspond to the indicated refinement levels r in OpenMP on a multicore system.

	Refinement levels (r)	No of Mesh Elements (N_e)	Number of degrees of freedom (DOF)	Error	1 Thread	2 Threads	3 Threads	4 Threads
Drug Release	1	5,388	13,950	1e-082	55s	42s	37s	31s
	2	21,552	54,835	1e-079	206s	153s	136s	114s
	3	86,208	217,425	1e-076	1064s	832s	731s	624s
Drug Delivery	1	54,337	43,652	1e-7.015	686s	490s	433s	367s
	2	171,938	133,912	1e-6.699	4258s	3300s	2894s	2475s
	3	551,283	413,320	1e-5.679	8020s	6120s	5359s	4590s

The message passing model for LINDA parallel program is well suited to predict the physical-chemical properties of benzene molecular structures. The speedup and effectiveness are increased linearly respect to a number of processors as shown in Figure-17.
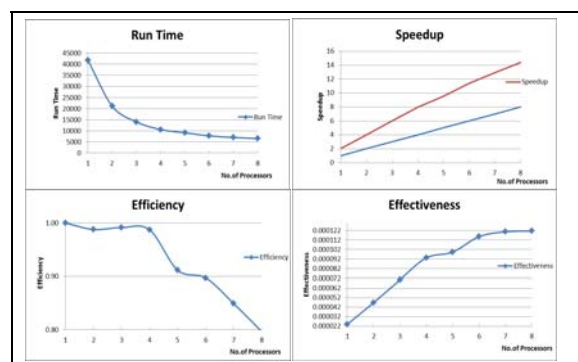

Figure-17. LINDA performance on multicore systems for predicting the chemical properties of benzene using MP4 method.

From Figure-2, the parallel performance evaluations of the improved Stefan problem coupled with the Navier–Stokes equations using 2 numerical schemes GSRB and AGE are shown in Figure-18. AGE is the superior scheme for solving the large sparse linear systems.
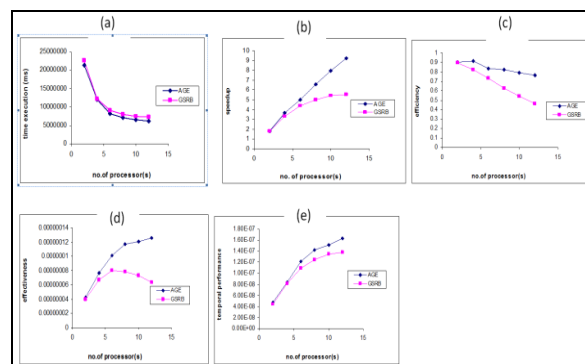
Figure-18. The parallel performance of improved Stefan problem coupled with the Navier–stokes equations for tire tread curing and molding processes in PVM based on two numerical schemes, GSRB and AGE.
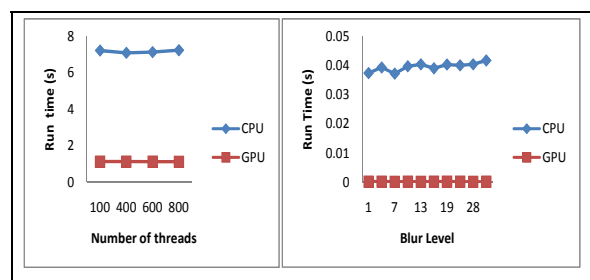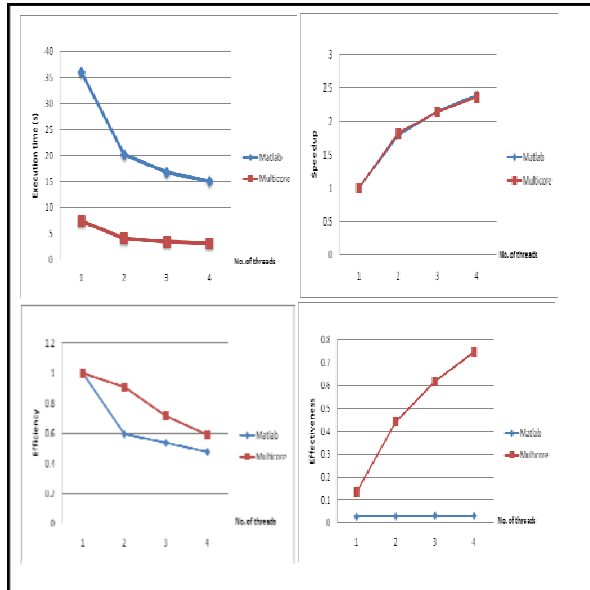

Figure-19. The GPU performance in CUDA on 1200 thread block and the blur level of image processing in Figure-3.
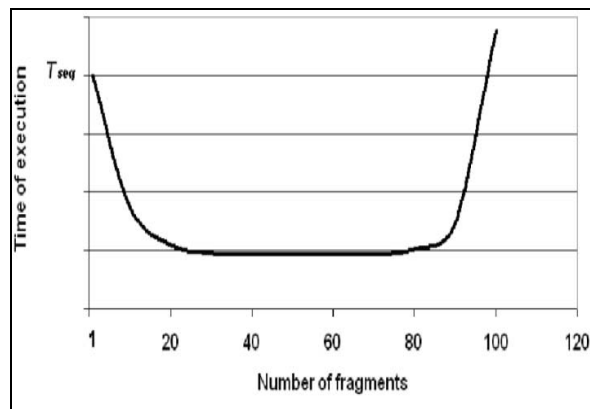
The comparison of parallel performance in Figure-20 shows that OpenMP is better than MDC for solving PDE model due to the communication model aspect.
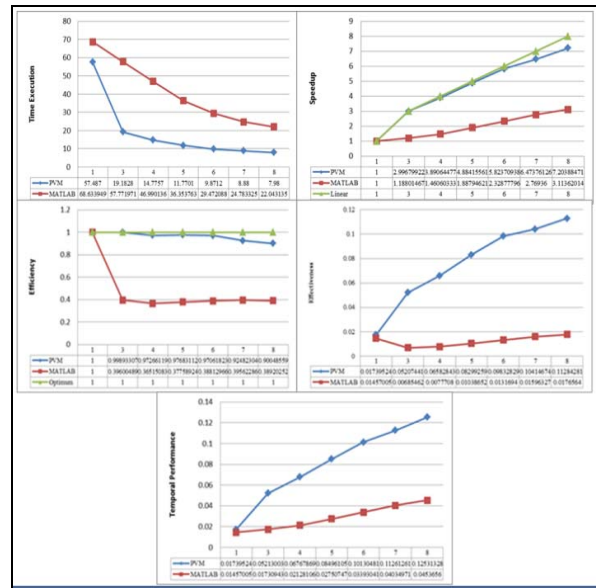

Figure-20. Performance comparison between OpenMP and MDC for solving PDE model in Figure 4.

Based on the performance in Figure-21, LUNA has statutes of limitation due to the communication model. The model is function between 20-90 numbers of fragments


Figure-21. The time of execution of fragmented program depending on the number of fragments relative to the time of execution of sequential program in LUNA.

The comparison of parallel performance in Figure 22 shows that PVM is better than MDC due to the communication model aspect.

Figure-22. The comparison of parallel performance evaluations between PVM and MDPC for solving the brain tumor simulation as stated in Figure-5.
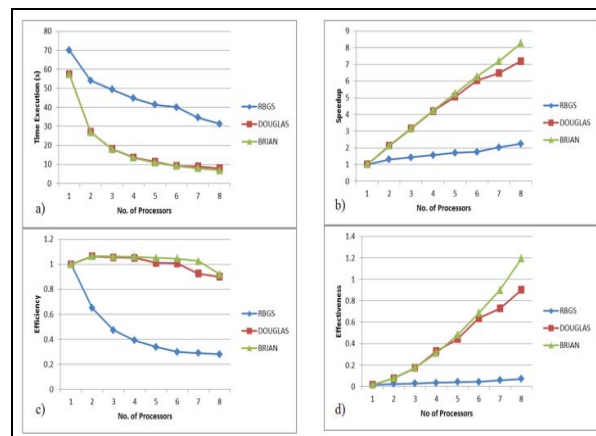
Figure-23. The comparison of parallel performance evaluations of some numerical schemes for solving the temperature behavior as stated in Figure-7.

CONCLUSIONS

The paper focuses on the design and development of parallel algorithms for the complex interdisciplinary problem involving a large sparse mathematical modeling. The selected communication models based on the parallel programming language are highlighted to support the high-speed simulation on HPC platform. The choice of communication models for a specific parallel program, shared and distributed memory classified as priority needs for solving the complex modeling. Communication costs become a critical issue to design and build the HPC software dealing with coarse and fine grained parallelism, communication activities and its design. By matching the communication model and the complex system, it will give an alternative strategy to produce an efficient algorithm for modeling the large sparse application,

intersection of a big data onto dynamically fine grids, clustering the grid points in a subdomain of a certain region according to specific criteria and structuring communication activities. As a conclusion, the integrated of HPC, communication model, PC software, parallel performance and numerical analysis are important strategies to fulfill the matching requirement and optimize the solution of complex interdisciplinary problems.

ACKNOWLEDGEMENTS

REFERENCES

[1] Nissani, M. 1997. Ten cheers for interdisciplinarity: The case for interdisciplinary knowledge and research. The social science journal, 34(2): 201-216.

[2] Klein, J.T. 1990. Interdisciplinarity: History, theory, and practice. Wayne state university press.

[3] Hamid, D.A., Islam, M.R., Alias, N. and Omar, A.H.H. 2010. An efficient authoring activities infrastructure design through grid portal technology. In Proc. of The 7th WSEAS International Conference on Engineering Education (EDUCATION'10), Corfu Island, Greece: 146-151.

[4] Alias, N. and Is, R. 2010. A Review of the Parallel Algorithms for Solving Multidimensional PDE Problems. Journal of Applied Sciences, 10: 2187-2197.

[5] Schlick, T. 2010. Molecular modeling and simulation: An interdisciplinary guide: an interdisciplinary guide (Vol. 21). Springer Science & Business Media.

[6] Sahnoun, R. and Mijoule, C. 2001. Density functional study of metal-arene compounds: Mono (benzene) chromium, bis (benzene) chromium and their cations. The Journal of Physical Chemistry A, 105(25): 6176-6181.

[7] Alias, N., Islam, M.R., Shahir, R., Hamzah, H., Satam, N., Safiza, Z., Darwis, R., Ludin, E. and Azami, M, 2010. Parallel system for abnormal cell growth prediction based on fast numerical simulation. In Russia-Taiwan Symposium on Methods and Tools of Parallel Processing. 222-231. Springer Berlin Heidelberg.

[8] Alias, N., Mustaffa, M.N., Saipol, H.F.S., Ghani, A. and Che, A. 2014. The Chronology of Geographic Information System Image Processing: A Case Study for Land Changes at Nusajaya, Iskandar Malaysia. Advanced Science Letters, 20(2): 435-438.

[9] Chamberlain, B.L., Callahan, D. and Zima, H.P. 2007. Parallel programmability and the chapel language. International Journal of High Performance Computing Applications, 21(3): 291-312.

[10] Callahan, D., Chamberlain, B.L. and Zima, H.P., 2004, April. The cascade high productivity language. In High-Level Parallel Programming Models and Supportive Environments, 2004. Proceedings. Ninth International Workshop on. 52-60. IEEE.

[11] Malyshkin, V. and Perepelkin, V., 2012. Optimization methods of parallel execution of numerical programs in the LuNA fragmented programming system. The Journal of Supercomputing, 61(1): 235-248.

[12] Mishra, S., Katiyar, V.K., Arora, V., Varshney, G. and Vishwavidyalaya, G.K. 2008. Mathematical Model of effect of drug delivery on blood flow in external magnetic field by Magnetic Nanoparticles. In Technical Proceedings Of The 2008 NSTI Nanotechnology Conference And Trade Show, NSTI-Nanotech, Nanotechnology. Boston, Massachusetts, USA. 45-48.

[13] Alias, N., Satam, N., Othman, M.S., Teh, C.R.C., Mustaffa, M.N. and Saipol, H.F. 2014. High Performance Nanotechnology Software (HPNS) for Parameter Characterization of Nanowire Fabrication and Nanochip System. In International Conference on Intelligent Software Methodologies, Tools, and Techniques. 251-268. Springer International Publishing.

[14] Alias, N, Bajuri, N.K, Mahmud, N.A. 2014. Ice Sheet Behaviour and Its Visualization. Journal of Applied Environmental and Biological Sciences. 8(1): 551-559

[15] Alias, N., Mokhtar, H., San, Y., Saipol, H.F.S., Mustaffa, M.N., Said, N.M. and Nor, N.M., 2014. High Speed Computation of Muscle Stress Problem Using Crack Propagation Modelling on DPCS Platform. Advanced Science Letters, 20(10-11): 1961-1966.

[16] Amdahl, G.M. 1967, April. Validity of the single processor approach to achieving large scale computing capabilities. In Proceedings of the April 18-20, 1967, spring joint computer conference. 483-485. ACM.