International Journal of Technology (2018) 1: 150-158
ISSN 2086-9614

THE PERFORMANCE OF A MOLECULAR DYNAMICS SIMULATION FOR THE *PLASMODIUM FALCIPARUM* ENOYL-ACYL CARRIER-PROTEIN REDUCTASE ENZYME USING AMBER AND GTX 780 AND 970 DOUBLE GRAPHICAL PROCESSING UNITS

Heru Suhartanto[1*], Arry Yanuar[2], Ari Wibisono[1], Denny Hermawan[1,3], Alhadi Bustamam[4]

[1] *Faculty of Computer Science, Universitas Indonesia, Kampus UI Depok, Depok 16424, Indonesia*
[2]*Faculy of Pharmacy, Universitas Indonesia, Kampus UI Depok, Depok 16424, Indonesia*
[3]*Faculty of Science and Technology, Al Azhar Indonesia University, Komplek Masjid Agung Al Azhar Jakarta 12110, Indonesia*
[4]*Faculty of Mathematics and Natural Sciences, Universitas Indonesia, Kampus UI Depok, Depok 16424, Indonesia*

(Received: December 2016 / Revised: March 2017 / Accepted: October 2017)

ABSTRACT

The invention of graphical processing units (GPUs) has significantly improved the speed of long processes used in molecular dynamics (MD) to search for drug candidates to treat diseases, such as malaria. Previous work using a single GTX GPU showed considerable improvement compared to GPUs run in a cluster environment. In the current work, AMBER and dual GTX 780 and 970 GPUs were used to run an MD simulation on the *Plasmodium falciparum* enoyl-acyl carrier protein reductase enzyme; the results showed that performance was improved, particularly for molecules with a large number of atoms using single GPU.

Keywords: GPU; Molecular dynamic simulation; PfENR

1. INTRODUCTION

Morris (1989) successfully created a computing-based drug design for antiretroviral raltegravir for HIV-1 using AutoDock, while other successful productions have included schistosomiasis drugs using InvDock (Suhartanto et al., 2012; Suhartanto et al., 2014). Molecular dynamics (MD) plays an important role in such drug-design activities and allows analysis of protein structures, protein folding, and molecular structures and properties (Amisaki & Fujiwara, 2004; Alam et al., 2008). MD simulations provide information about a single atom's position as a function of time, and during simulations, the position and velocity of all atoms are evaluated iteratively based on forces that act on each particle and Newton's laws of motion. The physical force or force field of each atom is calculated using three force components: bonded interaction, van der Waals, and electrostatic forces. Most MD simulations are time-consuming processes, and high-performance computing infrastructures with multi-core systems and GPUs have been used to reduce the time required to run simulations (Alam et al., 2008; Amaro & Li, 2010; Suhartanto et al., 2011; Suhartanto et al., 2012; Suhartanto et al., 2014; Liu et al., 2015). Previously, the current researchers used a single GTX GPU, which provided improved performance compared to GPUs in a cluster environment (Suhartanto et al., 2012; Suhartanto et al., 2014).

Quinine, primaquine, chloroquine, pyrimethamine, and sulfadoxine are antimalarial drugs that

*Corresponding author's email: heru@cs.ui.ac.id, Tel: +62-21- 7863419, Fax: +62-21- 7863419
Permalink/DOI: https://dx.doi.org/10.14716/ijtech.v9i1.1186

are widely used, although the *Plasmodium falciparum* parasite is resistant to existing antimalarial drugs due to spontaneous mutations in the structure and activity of drug targets in malaria parasites. Thus, it is very important to develop antimalarial drugs that specifically target parasites, such as *Plasmodium falciparum* enoyl-acyl carrier protein reductase (*Pf*ENR), plasmepsin (PM), and farnesyl transferase (FTase). In the past decade, *Pf*ENR was identified as a potential target for antimalarial drugs because it is found in a pathway of the type-II fatty acid biosynthesis known to take place in *Plasmodium falciparum* (Tasdemir, 2006; Frecer et al., 2009), where it plays an important role in catalyzing the final step in the elongation cycle of fatty acid biosynthesis. *Pf*ENR reduces the carbon double bond in enoyl that is covalently bound to the acyl carrier protein (Chhibber et al., 2006; Ben Mamoun et al., 2009; Agarwal & Fishwick, 2010).

Researchers have also accelerated computation time using AMBER, which is a collaborative framework, and the Tianhe-2 supercomputer (Peng et al., 2016), which achieves computational speeds 25–33 times faster than the sequential process. In addition, Intel Xion Phi multiple integrated circuit coprocessors have been used to accelerate explicit solvent implementations in the AMBER application package, resulting in speeds 4.17 times faster than the sequential process. The current work seeks to improve the performance of MD simulations using AMBER in a GPU-computing environment to process the conformational ensemble of *Pf*ENR to build a drug candidate for malaria. Dual GTX 780 and 970 GPUs were used to run the MD simulation, which is usually performed using cluster-computing resources, depending on the number of processors, amount of memory, and the network between the nodes in a cluster. Network communication should be reliable between the nodes; thus, networks are built based on fiber optic cables. However, MD is now performed in GPU environments, and one NVIDIA GTX 970 GPU can contain 1,664 cores and 4,096 MB of memory. When used for MD simulations, this decreases overhead communication between the nodes, which is what occurs in cluster environments.

2. EXPERIMENTAL

This work combined results from the literature to prepare a GPU-computing environment, compounds, simulations, and molecular modeling. MD simulations applying AMBER (Case et al., 2005), CHARMM (MacKerell et al., 1998), and OPLS (Jorgensen et al., 1996) as force fields for all atom simulations that model potential energy functions can be written as follows:

$$V(r) = \sum_{bonds} k_b(b - b_0)^2 + \sum_{angles} k_\theta(\theta - \theta_0)^2 + \sum_{dihedrals} k_\phi(1 + \cos(n\phi - \phi)) + \sum_{improper} k_\psi(\psi - \psi_0)^2 + \sum_{\substack{non-bonded \\ pairs\ (i,j)}} 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \sum_{\substack{non-bonded \\ pairs\ (i,j)}} \frac{q_i q_j}{\varepsilon_D r_{ij}} \quad (1)$$

where V is the potential energy function; r is the position vector for all particles; b, r_{ij} is the inter-particle distance; θ is the bond angle; ϕ is the dihedral angle; ψ is the improper dihedral angle; k_b, k_θ, k_ϕ, and k_ψ are the respective force constants; and ε_D is the dielectric constant.

Figure 1 illustrates the AMBER workflow beginning with the preparation of the PDB file, which can be obtained from the Protein Data Bank. Then, a topology file (prmtop) and coordinate file (Inpcrd) are produced using the control data file (mdin), and the experiment is produced using particle mesh Ewald molecular dynamics, which provides trajectory files (mdout and rstrt).

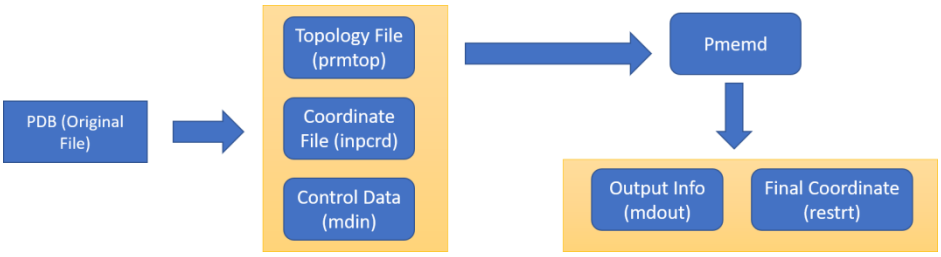

Figure 1 AMBER workflow (system model)

2.1. Preparation of Computing Environments

A Future Grid computing environment or cluster delta (Future Systems, Digital Science Center, School of Informatics and Computing, Indiana University, 2015), XSEDE or cluster stampede (XSEDE, 2011), and PC research computing environment were used to determine if computing environments are based on standard cloud computing from the National Institute of Standards and Technology of the US Department of Commerce (Mell & Grance, 2012). The characteristics of the computing environments used are listed in Table 1 (Suhartanto et al., 2012).

Table 1 Computing environment characteristics

Cloud-computing Characteristic	Future Grid	XSEDE	PC Research
On-demand self-service	V	V	X
Broad network access	V	V	V
Resource pooling	V	V	X
Rapid elasticity	V	V	X
Measured service	V	V	X

Table 1 shows that both Future Grid and XSEDE had the five characteristics of cloud computing, while PC research did not. Future Grid and XSEDE are platform as service systems, in which users run applications without needing to prepare necessary infrastructures, such as networks, servers, storage, or operating systems.

2.2.1. Preparation of a future grid delta cluster

Future Grid provides high-performance computing environments that allow researchers to run complex computations on cloud infrastructure. In this work, the computing environment used delta machines, which are GPU clusters with 16 nodes, 32 CPUs, and 192 cores running on Red Hat Linux. Detailed specifications for the Future Grid delta are provided in Table 2.

Table 2 Future grid delta specifications

Future Grid Delta Specifications
• 16 nodes and 192 cores
• Two Intel X5560 six-core 2.8 GHz processors per node
• 192 GB of DDR3 RAM
• Two NVIDIA Tesla C2070 GPUs with 448 cores per node

2.2.2. Cluster preparation of the XSEDE stampede

The second cloud-computing environment used an XSEDE Stampede Dell PowerEdge C8220 cluster consisting of 6,400 Dell DCS Zeus nodes, each of which had two Intel Xeon E5-2680 2.7 GHz processors (i.e., Sandy Bridge) with 32 GB of memory and 50 GB of storage per node; each

of the 128 nodes was attached to an NVIDIA Kepler GPU. Detailed specifications for XSEDE Stampede are given in Table 3.

Table 3 Specifications of the XSEDE stampede

XSEDE Stampede Specifications	
Host	stampede.tacc.xsede.org
Site	tacc.xsede.org
CPU Type	Intel Xeon E5-2680
Operating System	Linux (CentOS)
Processor Cores	102,400
Nodes	6,400
Memory	200 TB
Peak Performance	9,600 TFlops
Disk	14,336 TB
Local Storage Per Node	50 GB
Primary Storage Shared	14 PB
Storage Network	FDR InfiniBand
Interconnect	FDR InfiniBand
Batch System	SLURM
Startup Allocation Limit	50,000
Graphics Card	NVIDIA Kepler 2
Parallel File System	Lustre
Memory Per CPU	2 GB
CPU Speed	2.7 GHz
CPU Cores Per Node	16

2.2.3. Preparation of PC research

Both GTX 780 and GTX 970 GPUs, associated with PCI-E 2.0×16 slot lanes, were installed on the computer. Table 4 provides detailed specifications for the PC research.

Table 4 PC Research specifications

PC Research Specifications	
Operating System	Linux (Debian 3.2)
Processor Core	Intel Core I7-4790K Quad Core, Socket LGA1150
Memory	32 GB
Disk	2 TB, 250 GB
Graphics Card	2 x NVIDIA Geforce GTX 780
Socket	2x PCIe 16x

The details for the GPU used in the Future Grid Delta, XSEDE Stampede, and PC research are provided in Table 5. These details can be obtained by running /deviceQuery in each of the machines and are also available from the LLC (2012). The power board data are available at the NVIDIA 4 website. In Table 5, the Tesla K20m and GTX 780 GPUs had the same CUDA capability version of 3.5, while the Tesla C2070 GPU had version 2.0. The CUDA cores for the Tesla K20m, GTX 780, and Tesla C2070 GPUs were 2,496, 2,304, and 448, respectively, and the clock-rate memory of the GTX 780 GPU was the best. For memory bus width, the Tesla C2070 and GTX 780 GPUs were 384 bit, while the Tesla K20m GPU was 320 bit.

Table 5 Specifications of the K20m, TeslaC2070, GTX780, and GTX 970 GPUs

Specifications	Tesla K20m	TeslaC2070	GTX 780	GTX970
CUDA Capability Version Number	3.5	2.0	3.5	5.2
Total Amount of Global Memory	4,800 MB	5,375 MB	3,072 MB	4,095 MB
Multiprocessors CUDA Cores/MP CUDA Cores	13; 192; 2496	14; 32; 448	12; 192; 2304	13; 128; 1664
GPU Clock Rate	706 MHz	575 MHz	941 MHz	1,253 MHz
Memory Clock Rate	2,600 Mhz	1,494 Mhz	3,004 Mhz	3,505 MHz
Memory Bus Width	320 bit	384 bit	384 bit	256 bit
L2 Cache Size	1,310,720 bytes	786,432 bytes	1,572,864 bytes	1,835,008 bytes
Total Constant Memory	65,536 bytes	65,536 bytes	65,536 bytes	65,536 bytes
Total Shared Memory per Block	49,152 bytes	49,152 bytes	49,152 bytes	49,152 bytes
Total Available Registers per Block	65,536	32,768	65,536	65,536
Warp Size	32	32	32	32
Maximum Threads per Block	1,024	1,024	1,024	1,024
Max Dimension of Thread Block (x, y, z)	(1,024, 1,024, 64)	(1,024, 1,024, 64)	(1,024, 1,024, 64)	(1,024, 1,024, 64)
Max Dimension of Grid (x, y, z)	(2,147,483,647, 65,535, 65,535)	(65,535, 65,535, 65,535)	(2,147,483,647, 65,535, 65,535)	(2,147,483,647, 65,535, 65,535)
Board Power	225 W	247 W	250 W	145 W

2.3. Preparation of Input Data

The macromolecule data used as inputs for the study were taken from the Protein Data Bank for *Pf*ENR (id. 3LT0). *Pf*ENR is a specific target for antimalarial drug discovery, and examples from AMBER proteins located in the directory `AMBERHOME/test/CUDA` after compilation and the installation process were also used, namely TRPCage (304 atoms), myoglobin (2,495 atoms), and JAC (23,558 atoms). Macromolecular structures of *Pf*ENR targets were sought from the Protein Data Bank website, and macromolecules were selected based on inclusion criteria, such as being a wild-type or non-mutant macromolecule or related ligand. The exclusion criteria were resolutions greater than 2.5 Å and incomplete chains. The selected macromolecules were downloaded in text format then in PDB format for further processing.

3. RESULTS

The MD simulations were performed on four proteins: *Pf*ENR (37,873 atoms), JAC (23,558 atoms), TRPCage (304 atoms), and myoglobin (2,492 atoms). Five simulations were performed with time steps of 100, 200, 300, 400, and 500 pico seconds (ps) on Future Grid XSEDE and PC research using GTX 780 and GTX 970 GPUs. Average performance was measured in ns/day. Simulations using the GTX 780 or two GTX 780 GPUs were denoted as GTX980D and using the GTX 970 or two GTX 970 GPUs were denoted as GTX 970D, with time steps ranging from 21.32–21.39 ns/day (Figure 2). In comparison, simulations in Future Grid using the Tesla C2070 GPU had time steps ranging from 8.26–8.37 ns/day.

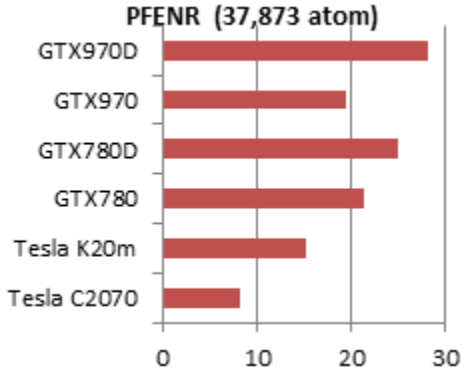

Figure 2 Comparison of GPU average performance (ns/day) for the *Pf*ENR protein

Figure 3 shows that the simulations using GTX 970D performed best followed by simulations using the GTX 980 GPU. The Tesla C2070 GPU was 1.5 times slower than the Tesla K20m and 3.0 times slower than the GTX970D.

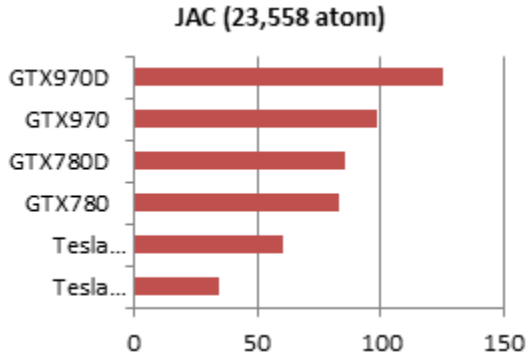

Figure 3 Comparison of GPU average performance (ns/day) for the JAC protein

Figure 4 shows the superior performance of the Tesla C2070 GPU compared to the other types, while neither the double GTX 780 nor GTX 970 GPUs performed better compared to the single GPUs due to the small sizes of and size differences among the proteins.

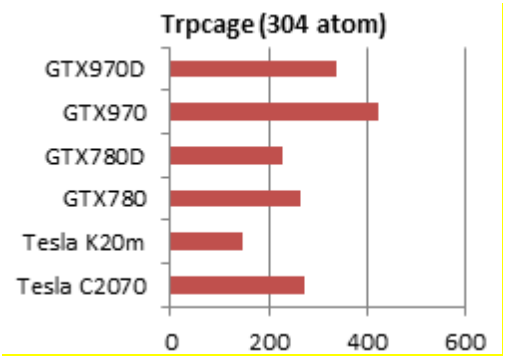

Figure 4 Comparison of GPU average performance (ns/day) for the TRPCage protein

Figure 5 shows that the Tesla C2070 GPU performed poorly compared to the Tesla K20m, GT780, double GTX 780, GTX 970, and double GTX 970 GPUs. The Tesla C2070 GPU has 5,375 MB of memory, which was the largest among the Tesla K20m (4,800 MB), GTX 780 (3,072 MB), and GTX 970 (4,095 MB) GPUs. As shown in Table 6, the simulations for all three proteins did not utilize 100% of the memory available.

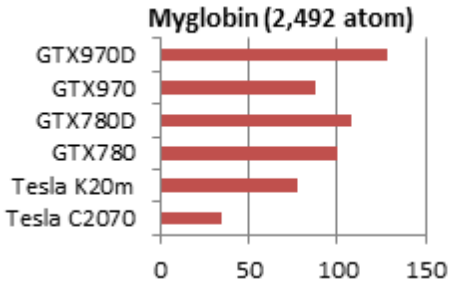

Figure 5 Comparison of simulation performance for myoglobin in ns/day

Table 6 GPU and CPU memory usage for the entire experiment

Protein	GPU Memory Use (KB)	CPU Memory Use (KB)
PfENR	214,043	42,755
JAC	60,507	24,219
Myoglobin	1,679	1,679
TRPcage	213	213

Table 6 shows the approximate memory use for each protein simulation. The figures were obtained from the estimates in the AMBER output file (i.e., mdout). Memory usage when running simulations were obtained using the command - nvidia- smi.

4. CONCLUSION

The results show that GPU specifications can be used to determine the performance of MD simulations, with the number of cores and memory clock rate being primary factors affecting simulation performance. Using AMBER and two GPUs for large proteins, such as *Pf*ENR (37,873 atoms) which is a potential target for antimalarial drugs, JAC (23,558 atoms), and myoglobin (2,492 atoms), can improve performance in ns/day by two-fold and reduce execution time compared to using a single GPU. For small proteins, such as TRP cage (304 atoms), using two GPUs does not improve performance. A significant improvement in simulation performance occurred for molecules that had more atoms because, to communicate between GPUs, the hardware must synchronize data from the first GPU, the CPU memory, the CPU, and the second GPU. If the amount of data is small, there will be overhead to this synchronization, but if the amount of data is large the synchronization process is more efficient, which improves simulation performance by reducing simulation time. In addition to inputting the size of the protein, the size of the threads per block also affects performance as measured by ns/day. Thus, a maximum thread value per block based on GPU specifications should be used to improve simulation performance.

MD simulations can be performed in a public cloud-computing environment, in which all infrastructure requirements for the hardware and software have been provided by the service provider, including support tools to simplify the user interface. This is very helpful for researchers who do not have the ability to build the infrastructure, install the hardware or software, or configure system administration. Additionally, users do not have to solve problems in the infrastructure, such as power or cooling issues. Nevertheless, the hardware provided by the service provider, as well as the waiting time prior to execution, depends on the level of activity in the computing environment, which also affects the overall simulation time.

5. REFERENCES

Agarwal, A.K., Fishwick, C.W.G., 2010. Structure-based Design of Anti-infectives. *Annals of the New York Academy of Sciences*, Volume 1213(1), pp. 20–45

Alam, S.R., Agarwal, P.K., Hampton, S.S., Ong, H., 2008. Experimental Evaluation of Molecular Dynamics Simulations on Multi-core Systems. *In*: Proceedings of the 15th International Conference on High Performance Computing, pp. 616–622

Amaro, R., Li, W., 2010. Emerging Methods for Ensemble-based Virtual Screening. *Current Topics in Medicinal Chemistry*, Volume 10(1), pp. 3–13

Amisaki, T., Fujiwara, S., 2004. Grid-enabled Applications in Molecular Dynamics Simulations using a Cluster of Dedicated Computers. *In*: International Symposium on Applications and the Internet Workshops (Saint 2004Workshop). Tokyo, Japan, 26 January 2004, pp. 616–622

Ben Mamoun, C., Prigge, S.T., Vial, H., 2009. Targeting the Lipid Metabolic Pathways for the Treatment of Malaria. *Drug Development Research*, Volume 71(1), pp. 44–55

Case, D.A., Cheatham, T.E., Darden, T., Gohlke, H., Luo, R., Merz, K.M., Onufriev, A., Simmerling, C., Wang, B., Woods, R.J., 2005. The AMBER Biomolecular Simulation Programs. *Journal of Computational Chemistry*, Volume 26(16), pp. 1668–1688

Chhibber, M., Kumar, G., Parasuraman, P., Ramya, T.N.C., Surolia, N., Surolia, A., 2006. Novel Diphenyl Ethers: Design, Docking Studies, Synthesis and Inhibition of Enoyl ACP Reductase of *Plasmodium falciparum* and *Escherichia coli*, *Bioorganic & Medicinal Chemistry*, Volume 14(23), pp. 8086–8098

Frecer, V., Megnassan, E., Miertus, S., 2009. Design and *in silico* Screening of Combinatorial Library of Antimalarial Analogs of Triclosan Inhibiting *Plasmodium falciparum* Enoyl-acyl Carrier Protein Reductase. *European Journal of Medicinal Chemistry*, Volume 44(7), pp. 3009–3019

FutureSystems, 2015. *FutureSystems*. Available online at: http://futuregrid.org

Jorgensen, W.L., Maxwell, D.S., Tirado-Rives, J., 1996. Development and Testing of the OPLS All-atom Force Field on Conformational Energetics and Properties of Organic Liquids. *Journal of the American Chemical Society*, Volume 118(45), pp. 11225–11236

Liu, X., Peng, S., Yang, C., Wu, C., Wang, H, Cheng, Q, Zhu, W, Wang, J., 2015. mAMBER: Accelerating Explicit Solvent Molecular Dynamic with Intel Xeon Phi Many-Integrated Core Coprocessors. *In*: Proceedings of the 2015 IEEE/ACM 15th International Symposium on Cluster, Cloud, and Grid Computing, pp. 729–732

LLC, A.P.C., 2012. *New 4-day Training Course on GPU-enabled Neural Networks*

MacKerell, A.D., Bashford, D., Bellott, M., Dunbrack, R.L., Evanseck, J.D., Field, M.J., Fischer, S., Gao, J., Guo, H., Ha, S., Joseph-McCarthy, D., Kuchnir, L., Kuczera, K., Lau, F.T.K., Mattos, C., Michnick, S., Ngo, T., Nguyen, D.T., Prodhom, B., Reiher, W.E., Roux, B., Schlenkrich, M., Smith, J.C., Stote, R., Straub, J., Watanabe, M., Wiórkiewicz-Kuczera, J., Yin, D., Karplus, M., 1998. All-atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins. *Journal of Physical Chemistry B*, Volume 102(18), pp. 3586–3616

Mell, P., Grance, T., 2012. *The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology Special Publication 800-145*

Morris, G.M., 1989. *AutoDock's Role in Developing the First Clinically-approved HIV Integrase Inhibitor—AutoDock*

Peng, S., Zhang, X., Lu, Y., Liao, X.-K., Lu, K., Yang, C., Liu, J., Zhu, W., Wei, D.-Q., 2016. mAMBER: A CPU/MIC Collaborated Parallel Framework for AMBER on Tianhe-2 Supercomputer. *In*: Proceeding of IEEE International Conference on Bioinformatics and Biomedicine, BIBM, Shenzhen, China, December 15-18, 2016, pp. 651–657

Suhartanto, H., Yanuar, A., Bustamam, A., Azizah, A.Y., Wibisono, A., Hilman, M., 2014. Performance Analysis of Molecular Dynamics Simulation of PfENR Enzyme using AMBER on Cluster and GPU Computing Environment. *International Journal of Advancements in Computing Technology*, Volume 6(1), pp. 68–78

Suhartanto, H., Yanuar, A., Hilman, M., Wibisono, A., Dermawan, T., 2012. Performance Analysis, Cluster Computing Environments on Molecular Dynamics Simulation of LOX-RAD GTPase and Curcumin Molecules with AMBER. *International Journal of Computer Science Issues*, Volume 9(2), pp. 90-96

Suhartanto, H., Yanuar, A., Wibisono, A., 2011. Performance Analysis Cluster and GPU Computing Environment on Molecular Dynamic Simulation of BRV-1 and REM2 with GROMAC. *International Journal of Computer Science Issues*, Volume 8(4), pp. 131-135

Tasdemir, D., 2006. Type II Fatty Acid Biosynthesis, a New Approach in Antimalarial Natural Product Discovery. *Phytochemistry Reviews*, Volume 5(1), pp. 99–108

XSEDE, 2011. *XSEDE*. Available online at: https://www.xsede.org/