THE DEVELOPMENT OF INTEGRATED COMPUTING PLATFORM TO IMPROVE USER SATISFACTION AND COST EFFICIENCY OF IN SILICO DRUG DISCOVERY ACTIVITIES

[*1]Heru Suhartanto, [2]Xue Li, [3]Kevin Burrage, [4]Arry Yanuar, [5]Alhadi Bustamam, [6]Muhammad Hilman, and [7]Ari Wibisono

*[*1, corresponding author]Faculty of Computer Science, Universitas Indonesia, heru@cs.ui.ac.id*
[2]School of ITEE, the University of Queensland, xueli@itee.uq.edu.au
[3]Oxford University, kevin.burrage@gmail.com
[4]Faculty of Pharmacy, Universitas Indonesia, arry.yanuar@ui.ac.id
[5]Faculty of Math and Natural Science, Universitas Indonesia, alhadi@sci.ui.ac.id
[6,7]Faculty of Computer Science, Universitas Indonesia, h.hilman@cs.ui.ac.id, ari.wibisonozbw@gmail.com

Abstract

The research on drug discovery has a long history. The research quality can be improved by using information technology as it can be used to model many aspects inside the process. Molecular dynamics is the first stage of the drug discovery process that models the finding of best conformation protein structure. The chosen structure is then used as a receptor for several datasets of ligand in virtual screening process. The main goal of this paper is developing an integrated computing platform for drug discovery research using "as a service" paradigm. Here, the cloud applications is used as a bridge between drug discovery tools that running on transparent multiplatform clients. This paper reports the results of the first from three phases of the research - the development of the computing platform interfaces and the analysis of several computing resources including the commercial cloud that will be used by the system.

Keywords: *drug discovery, cloud computing, molecular dynamics, virtual screening*

1. Introduction

Molecular dynamics is considered at the first stage of drug discovery process. Molecular dynamics simulation is used for protein modeling. A protein in this case could be a virus or bacteria that simulated using the inhibitor (a protein that did prevention for another protein to develop) with various degrees of temperature and pressure. Investigations on molecular dynamics must be performed to understand the formations, structures and interactions among molecules. This is required in order to select the best conformation of molecules. The output of molecular dynamics then is used for virtual screening of substance datasets that are presumed to be the candidate of a medicine.

Molecular dynamics and virtual screening consist of several computations and numerical modeling that consume a huge computational resource. Due to huge number of atoms in molecules, it takes a long time to run the simulation. However, the advancement of information technology has developed many concepts, theories and tools that enhances the process and also the quality of this activity. The concept of parallel computing with its derivation is one of the solutions of this huge resource research problem. Currently, simulations can be performed much faster using either cluster computers or machines with GPUs.

These molecular dynamics and virtual screening tools use script, terminal, and some of them are already equipped with GUI (Graphic User Interface). The applications have several stages to complete all of the operations. In each stage of the simulations, users are asked to complete some of the parameters which are used to perform simulations. Some applications still use a console based interface to perform the simulation.

As the users are mostly researchers that are not computer scientists, all steps that must be done in the console is quite troublesome. Another problem lies on the setup and configuration of the computing environment, such as infrastructure development, system installation, and application

configuration surely takes a lot of effort and resources. Therefore, in this research, we propose a model of cloud applications where the users of molecular dynamics and virtual screening can easily and quickly perform the steps in preparation, configuration, and simulation without any technical problems. We develop the tools to provide a service on cloud computing technology to improve the user's satisfaction and simplify the stages of doing research on molecular dynamics and virtual screening. We also study how this cloud prototype differs in the performance with that available in commercial services.

2. Drug Discovery Processes

2.1. Molecular Dynamics

Molecular dynamics method was first introduced by Alder and Wainwright in 1950. It was used to study the interaction of the hard balls (hard spheres) [1], the field of molecular dynamics simulations began developing in 1964, who named Rahman, the first Researchers which simulate fluid anbrgon, and in 1974 Stillinger Rahman try to simulate molecular dynamics in a realistic state of the system [2]. In 1977 protein simulation was first performed by McCammon using a pancreatic protein trypsin inhibitor [3]. Molecular dynamics simulation is widely use to study the dissolution of proteins, study of ligand binding, DNA-protein complex interactions, and protein folding.

The basic process of molecular dynamics is to determine the total force in the N atom system, acceleration of each atom, and the speed of each atom. After the atoms interact with each others, the atoms will move into new positions [4].

Computing resources can help researchers to understand the microscopic properties of molecules and structures with the help of molecular dynamics simulations. The main objective in conducting molecular dynamics simulations is to produce a trajectory of molecules within a given time period. Each of the molecules in molecular dynamics simulations has the momentum and position parameters in period of time. So the expert can see the molecule transformations in each unit of time.

2.2. Virtual Screening

Molecular docking is a computational procedure that attempts to predict noncovalent binding of macromolecules. The goal is to predict the bound conformations and the binding affinity [5]. The prediction process is based on information that is embedded inside the chemical bond of substance.

Virtual Screening is molecular docking process that involved large database of chemical compound. We can say that virtual screening is a lot of molecular docking process to get the best molecule to become drug candidate. It is important to say that this simulation is not the only one method to get the drug candidates, practical laboratory experiment is the one that usually does that. This virtual experiment used to process large number of substance to produce small number of best drug candidates which cannot be done manually due to large number of data. The key to the parallelization process in molecular docking lies on this aspect, "big number of molecular docking jobs". We have performed this virtual screening project on clusters of computers; the speedup is linear, which is good, since this case can be classified as embarrassingly parallel problem [6].

3. Integrated Drug Discovery Platform

The authors propose a big picture of drug discovery cloud computing platform. The approach will provide services that serve each of the activities in this drug discovery processes. The idea is to integrate all activities and the resources into one computing infrastructure, thus providing convenient molecular modeling and virtual screening processes. This proposed platform consists of four main entities, namely molecular database; computing resources; drug design system, and user interface. Figure 1. describes the proposed system.

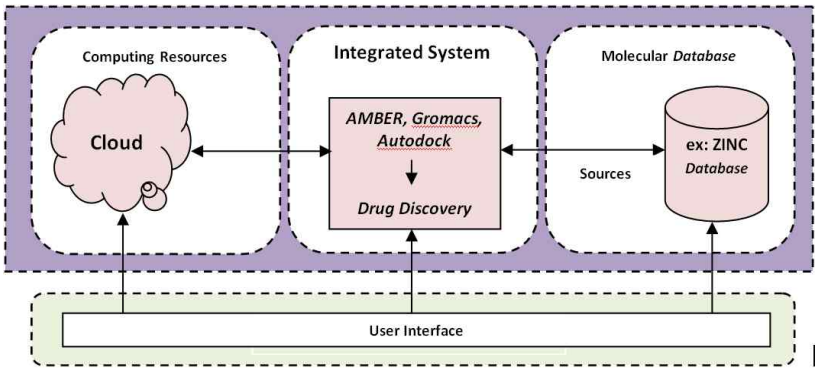

Figure 1. The design of integrated drug design system

A molecule database is the data source for virtual screening. There are some issues related to molecular database development that relate to the data standard format and data storage structure. This database stores well-annotated data for further usage. The biggest molecule database is ZINC – developed by the University of California, San Francisco – it consists of 14 million molecules that are ready to be processed further [7].

The second entity is the computational resources. Data processing in virtual screening needs high performance computing resources. Access to this resources in the form of cluster, grid or cloud computing is a must in drug discovery. Thus management of the resources will be part of the whole processes.

The third entity that is the core of drug design is an integrated drug design system. These entities consists of two process, namely molecular dynamics that studies the characteristics of protein molecule of the test object and virtual screening that searches for drug candidates appropriate for a certain protein molecule. It is a two communication processes between the database and drug design systems where the system take inputs and provide outputs to the database. There is also interaction between drug design systems with computing resources that is part of computing resource management process.

The last entity is the interface between user and the system. Requirement for user friendly interfaces are necessary for people whose interests are on molecules, chemistry, and pharmacy.

In this paper, we are working on the first step in developing the integrated system as seen in Figure 2. The foundation we build is in the computational resources and interfaces. This prototype resembles the second and forth entities of the integrated system and a sample of the third entity, which is molecular dynamics and virtual screening process.

Working in the first phase including the development of computing platform user interfaces that will be the first gate for the users to drug discovery tools and also the study of the effectiveness in using commercial cloud facility compared to on premises infrastructure as part of the computational resource.

4. SCloud Systems

We developed the interfaces for this system – called SCloud [8]. This interface is built based on the cloud computing principal model that provides several services for the user such as on-demand self services, broad network access, resource pooling, rapid elasticity, and measured services. We have developed this system and integrate two molecular dynamics applications, which are Gromacs and Amber, as we've experimented by those applications using cluster computer and GPU [9-10].

4.1. Network Design

Network design that is used to perform modeling and experiments is a LAN connection. These computers are connected using a Local Area Network with a simple topology as shown in Figure 3.

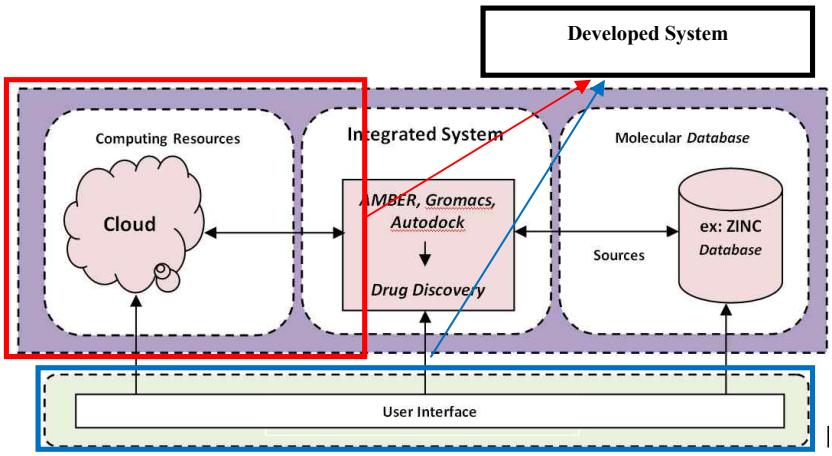

Figure 2. Computing platform and interfaces

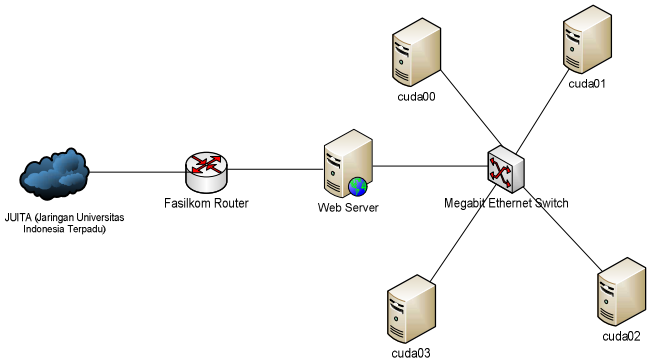

Figure 3. Network Design

Our proposed network design is divided into two categories:

1. Web Server

The cloud application is going to deploy in the web server and stores the script order to the worker node. Web server application will manage the creation of application and configuration of cloud application.

2. Computing Cluster

The computing cluster is a group of worker nodes to run job simulations. These worker nodes administered by direct order from cloud application in the web server. The installed application Gromacs and Amber are installed in this computing cluster. Computing cluster is also equipped with mounting storage that is shared over computing cluster and web server. Beside storage share, the computing cluster needs to be installed with MPI (Message Passing Interface) to perform parallel execution for Gromacs and Amber.

4.2. Software Design

Application that is installed in the web application server and computing nodes have some differences. In the web application server, this resource is not used as a resource that performs simulation. The web application server only stores the information and application configuration that is configured by the users. The web application server acts as an enabler for users to perform simulations using the computational resources in accordance with the user's requirements.

In this research, the cloud computing applications are made following the basic concept of the cloud characteristics. Figure 4 shows the general picture of cloud application use case design.

This system provides better experience for the users rather than using console or manual utility of the applications. There is no distortion of the stages and results in using this interface compared to the manual one. It has been tested by using several scenarios and the results shows that the output of the simulation is the same either with the developed system and the manual system. We also compared the usability SCloud system and manual system. We have tested the application involving ten people, every person gives the different result of the test. The parameters that we compared are the time of the feature use, the average use time of the system, the average mistake that is made by user, level of easiness for the user, level of satisfaction for the user.

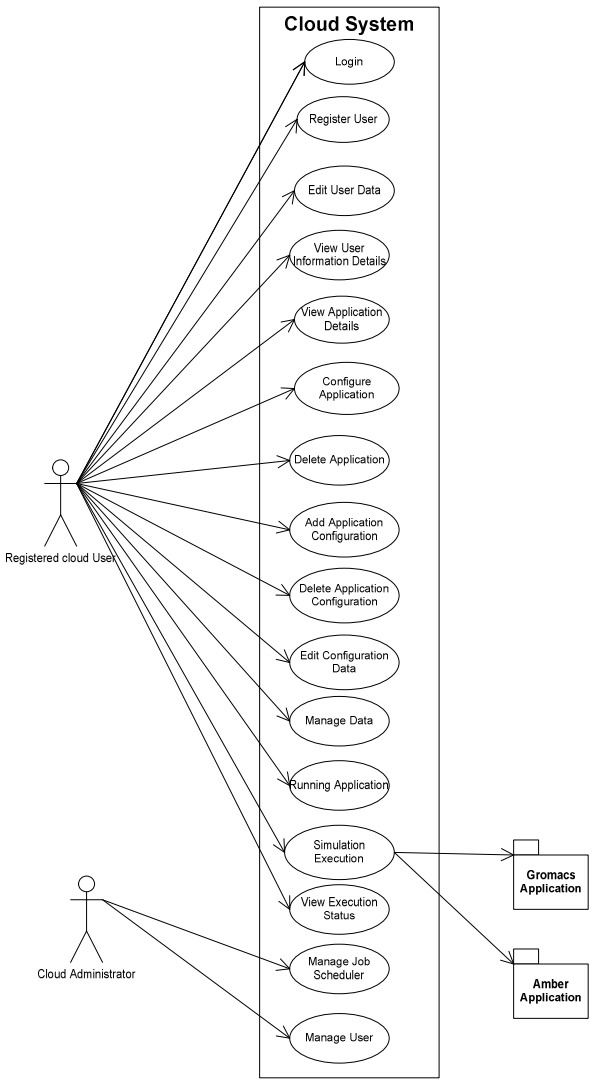

Figure 4. Use Case Design

Table 1. contains the parameter testing for the entire features that have been tested. All of the feature operation are 100% work, the average time of feature operation is 1569 second, the

average of user's mistakes is twice in each feature, and the level of easiness and user's satisfaction of the SCloud system is more than 4.5 of 5.

Table 1. SCloud System Testing

No	Parameter	Result
1	All Feature Operation	100 %
2	Average time of Feature Operation	1569 second
3	Average of user's mistake	Twice on each features
4	Level of easiness	4.92 of 5
5	Level of user satisfaction	4.7 of 5

Table 2. Manual System Testing

No	Parameter	Result (second)
1	Average execution time of the user establishment	3600 (s)
2	Average time user configuration	1800 (s)
3	Average time of user's script writing	500 (s)
	Total time	5900 (s)

From Table 1 and Table 2 it is obvious that in term of the feature execution time of both systems, the Scloud system has less average time of feature execution than that of the manual system.

5. Performance Analysis of Computational Resources

In this section, we study the computational resource that will be the source of the computing for the integrated system. We analysis the performance of our own cluster infrastructure compared to three different type of cluster built on commercial cloud facilities Amazon EC2. Evaluation metrics used in this study is based on seven tests that consists of several parameters used in HPC Challenge [11], they are

1. HPL (High Performance Linpack)

Modification of the Linpack benchmark for measuring floating point execution rate in the linear equation (in TFLOPS/s). There are two criteria used in these tests generally consisting of two types, namely 'rate' and 'time'.

2. DGEMM (per processor)

Benchmark for measuring floating point execution rate in double precision matrix multiplication in a single processor (in GFlops/s). There are two criteria used in this test that is generally a measure of 'rate'.

3. PTRANS (per processor)

Benchmarks implementation to measure the matrix transpose of data transfer and communication between processors in parallel with MPI processes (in GB/s). There are two criteria used in this test namely 'rate' and 'time'.

4. STREAM (per thread)

Simple benchmark for measuring the available bandwidth in memory (in GB/s). There are two criteria used in this test namely 'star' and 'single' in which each criterion measures the following four operations: copy, scale, add, and triad. In general, this test measures the 'rate'.

5. Random Access (per processor)

Benchmark for measuring the rate of integer random updates on machine memory (in GUP/s). There are 10 criteria used in these tests that generally consist of two types, namely 'rate' and 'time'.

6. Bandwidth & Latency (per process)

Benchmarks to measure bandwidth, network capacity to transmit 2MB of data (in GB/s) and latency, the time required to send 8-byte of data between processors (in microseconds). There are ten criteria used in these tests that generally consist of two types, namely 'rate' and 'time'.

7. FFTE (per process)

Benchmarks to measure the distribution process on all processors using Discrete FFT test (in GFlops/s). There are four criteria used in these tests that generally consist of two types, namely 'rate' and 'time'.

The experiment is conducted by building a virtual cluster; we label them as SCluster, LCluster, and XLCluster. The performance of the three virtual clusters on Amazon EC2 then compared with our own cluster named as Cluster05 as it used by the SCloud systems. The specification of three virtual clusters and one on the local cluster can be seen in the Table 3.

Table 3. Specifications cluster of four scenarios

Parameter	SCluster	LCLuster	XLCluster	Cluster05
Memory	1.7GB	7.5GB	15GB	4GB
Compute Unit/Processors	1 1.2GHz	4 1.2GHz	8 1.2GHz	Quad Core 2.8GHz
Platform	32-bit	64-bit	64-bit	32-bit
Cost/Hour	$0.085	$0.34	$0.68	-

Table 4. Results of HPC Benchmark

Parameter	**SCluster**	**LCluster**	**XLCluster**	**Cluster05**
rate (TFlops/s)	6.6179×10^{-4}	1.44817×10^{-3}	3.05157×10^{-3}	9.55014×10^{-3}
time (s)	1.00963	0.463188	0.218958	0.0699641
star (GFlops/s)	0.275904	2.03071	4.30878	3.22242
single (GFlops/s)	0.315812	1.66391	4.52124	3.4535
rate (GB/s)	0.0199802	0.036743	0.959246	0.0636943
time (s)	0.00785705	0.024838	0.00160098	0.00065105
*rate*_copy_star (GB/s)	14.1946	17.3262	15.9788	6.5341
*rate*_scale_star (GB/s)	13.4435	17.4807	16.1767	6.50222
*rate*_add_star (GB/s)	14.2664	17.1028	15.768	7.1814
*rate*_triad_star (GB/s)	13.8689	16.2864	15.3358	7.05201
*rate*_copy_single (GB/s)	14.3394	18.5178	18.8296	18.5178
*rate*_scale_single (GB/s)	13.4756	18.5178	18.8296	17.5861
*rate*_add_single (GB/s)	14.2906	18.3557	18.5178	18.5178
*rate*_triad_single (GB/s)	13.899	17.4037	15.3919	18.0214
MPI_LCG_Time (s)	21.2204	6.0435	0.206168	0.055549
MPI_LCG CheckTime (s)	0.292799	0.06738	0.0121179	0.015765
MPI_LCG_GUP/s	0.0000988273	0.00034701	0.0101721	0.0377532
MPI_Time (s)	21.1397	6.39533	0.203144	0.0542559
MPI_CheckTime (s)	0.248488	0.0981891	0.0149088	0.0135739
MPI_GUP/s	0.0000992046	0.000327919	0.0103235	0.0386529
Star_LCG_GUP/s	0.0742527	0.201557	0.225982	0.0960679
Single_LCG_GUP/s	0.111292	0.369522	0.37558	0.0962522
Star_GUP/s	0.0976685	0.311434	0.217506	0.110843
Single_GUP/s	0.129233	0.312095	0.309112	0.134257
Max PingPong Latency (ms)	25.0587	9.62615	0.695388	0.461407
Min PingPong Latency (ms)	18.3818	7.98702	0.417233	0.420439
Avg PingPong Latency (ms)	21.7036	8.96181	0.577006	0.445727
Max PingPong Bandwidth (GB/s)	0.785995	1.48985	2.12316	7.79709
Min PingPong Bandwidth (GB/s)	0.640923	0.170722	0.962438	7.54701
Avg PingPong Bandwidth (GB/s)	0.739044	1.19802	1.59297	7.7374
Random Ring Latency (ms)	26.0078	9.3362	1.13274	0.599244
Random Ring Bandwidth (GB/s)	0.499507	0.198583	0.321133	1.43202
Natural Ring Latency (ms)	26.0106	8.79765	0.09673	0.596122
Natural Ring Bandwidth (GB/s)	0.491766	0.11922	0.328975	1.42019
*rate*_star (GFlops/s)	0.724842	1.88002	1.80513	1.3635
*rate*_single (GFlops/s)	0.0302832	1.91526	0.0400881	1.48668
*rate*_MPI (GFlops/s)	0.240565	1.0272	2.13208	3.16013
MPI_avgtime (s)	0.003107851	0.000724725	0.000346422	0.000228167

The seven sets of data obtained from the experiments are presented in the Table 4 to facilitate and understand the information contained, the data is not presented in the diagram/curve as the data generated is relatively sparse.

To see the significance of differences in the performance of the four clusters used in the study, we used the Friedman test with 4 test objects (columns) and 38 test criteria (rows) [12]. – The Friedman test- was conducted to test multiple classifiers on multiple datasets. One of the advantages of the Friedman test is that it does not depend on the distribution of the data used. The Friedman test consists of two parts, the first part of the object doing the test based on criteria that are measured and then test the null hypothesis with a formula to calculate the distribution. 38 criteria drawn from seven benchmark test with the assumption that all test parameters have the same weight on the performance of the system.

First, we convert the information in the Table 4 into the same table but contain the rank of four clusters in each parameter. From this information, we can see the average rank of each cluster. The converted table can be seen in the Table 5. As we can see from Table 5, the average rank for 38 parameters shows that XLCLuster takes the first position then followed by Cluster05. The third and fourth position are filled by LCluster dan SCluster respectively. Generally, based on the rank, the XLCluster outperform our own cluster Cluster05, but we have to do the significance test.

So, for the second step, we will check whether this average rank is significance enough to determine the difference of performance for each cluster. The Friedman test will be performed on the data from the experiment with the hypothesis

H_0: no significant differences between the cluster
H_1: there are significant differences between the cluster

The formula used to calculate the Friedman distribution is as follows

$$x_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \quad (1)$$

Where

x_F^2 : Friedman distribution
N : test criteria (rows)
k : test object (column)
R_j : test object rank

The formula to calculate the value of Friedman test (F_F) that distributed on F – distribution by degree of freedom $(k-1)$ and $(k-1)(N-1)$ is

$$F_F = \frac{(N-1)x_F^2}{N(k-1)-x_F^2} \quad (2)$$

If H_0 is rejected, then we proceed to check the critical difference (CD) between 4 clusters using

$$CD = q_\alpha\sqrt{\frac{k(k+1)}{6N}} \quad (3)$$

where

q_α : critical value (for α = 0.05) = 2.343

From Table 5, the Friedman distribution can be calculated using the formula above by inserting the value N = 38, k = 4, average rank *SCluster* = 3.68, LCluster = 2.57, XLCluster = 1.83, and Cluster05 = 1.92 into equation (1) and then produce the results as follows

$$x_F^2 = \frac{12(38)}{4(5)}\left[(3.68^2 + 2.57^2 + 1.83^2 + 1.92^2) - \frac{4(5)^2}{4}\right] = 49.76$$

Table 5. Average Rank of The Clusters

Parameter	SCluster	LCluster	XLCluster	Cluster05
rate (TFlops/s)	4	3	2	1
time (s)	4	3	2	1
star (GFlops/s)	4	3	1	2
single (GFlops/s)	4	3	1	2
rate (GB/s)	4	3	1	2
time (s)	3	4	2	1
*rate*_copy_star (GB/s)	3	1	2	4
*rate*_scale_star (GB/s)	3	1	2	4
*rate*_add_star (GB/s)	3	1	2	4
*rate*_triad_star (GB/s)	3	1	2	4
*rate*_copy_single (GB/s)	4	2.5	1	2.5
*rate*_scale_single (GB/s)	4	2	1	3
*rate*_add_single (GB/s)	4	3	1.5	1.5
*rate*_triad_single (GB/s)	4	2	3	1
MPI_LCG_Time (s)	4	3	2	1
MPI_LCG CheckTime (s)	4	3	1	2
MPI_LCG_GUP/s	4	3	2	1
MPI_Time (s)	4	3	2	1
MPI_CheckTime (s)	4	3	2	1
MPI_GUP/s	4	3	2	1
Star_LCG_GUP/s	3	2	1	4
Single_LCG_GUP/s	3	2	1	4
Star_GUP/s	4	1	2	3
Single_GUP/s	4	1	2	3
Max PingPong Latency (ms)	4	3	2	1
Min PingPong Latency (ms)	4	3	1	2
Avg PingPong Latency (ms)	4	3	2	1
Max PingPong Bandwith (GB/s)	4	3	2	1
Min PingPong Bandwith (GB/s)	3	4	2	1
Avg PingPong Bandwith (GB/s)	4	3	2	1
Random RingLatency (ms)	4	3	2	1
Random Ring Bandwith (GB/s)	2	4	3	1
Natural RingLatency (ms)	4	3	1	2
Natural Ring Bandwith (GB/s)	2	4	3	1
*rate*_star (GFlops/s)	4	1	2	3
*rate*_single (GFlops/s)	4	1	3	2
*rate*_MPI (GFlops/s)	4	3	2	1
MPI_avgtime (s)	4	3	2	1
Average rank	3.68	2.57	1.83	1.92

Then the Friedman value is calculated by inserting the Friedman distribution value obtained in the previous process in to the equation (2) and then produce the results as follows

$$F_F = \frac{(38-1)49.76}{38(3)-49.76} = 28.66$$

The critical value for F (3,111) with $\alpha = 0.05$ is 3.07. As $F_F = 28.66$ is greater than F (3,111) = 3.07, H_0 is rejected. Then, we calculate the CD value to see the critical difference of average rank of 4 clusters.

$$CD = 2.343\sqrt{\frac{4(5)}{6(38)}} = 0.694$$

We obtained the CD value = 0.694, which means that the performance of the clusters is considered different significantly if the difference between the average value is 0.694 or more. We can see that the difference between the average rank of the first rank Amazon EC2 cluster XLCLuster and our own

cluster Cluster05 is 0.09 which is below 0.694. This means the performance of both two highest rank cluster is not significantly different. Thus, we can consider both systems are in the same power in providing computational resources for drug discovery activities.

6. Conclusion

Cloud computing platforms are useful for drug discovery processes. Stages and phases in this research can be simplified using our proposed system. Our proposed system lets users to create elasticity of various stages and phases in doing the simulation. As it shows in the user acceptance test, all of the users are satisfied with the proposed system with very high rates. The improvement of user experience can be seen in the average time of the operations in the proposed system that only takes 1/3 of the times in the conventional one.

As with the computational resources, we can see that our own Cluster05 has fair performance compared to XLCluster from the amazon EC2. But, there are several things that will simplify the infrastructure establishment if we use the Amazon EC2 as a IaaS rather than setup on premises infrastructure for cluster computers. We will explore and analyze these matters in future work

7. Acknowledgment.

This work is partly funded by Kerjasama Luar Negeri Research Program, Directorate General of Higher Education, The Ministry of Education and Culture, Indonesia based contract No 2492/H2.R12/HKP.05.00/2013

8. References

[1] Alder, B. J., & Wainwright, T. E., "Phase Transition for a Hard Sphere System", J. Chem. Phys., 27, 1208, 1957
[2] Stillinger, F. H., & Rahman, A, "Improved Simulation of Liquid Water by Molecular Dynamics," J. Chem. Phys., 60, 1545-1557, 1974
[3] McCammon, J. A., Gelin, B. R., & Karplus. London: M. Nature.
[4] Yulianto, T.D., Analisis Kinerja Cluster Hastinapura Menggunakan Dinamika Molekular RAD GTPase dengan Aplikasi AMBER. Undergraduate Thesis, Universitas Indonesia, Faculty of Computer Science, Depok, 2010
[5] Trott, O. & Olson, A.J., "Autodock Vina : Improving The Speed and Accuracy of Docking with a New Scoring Function, Efficient, Optimization, and Mutithreading" (Journal Online Sources Style), Journal of Computational Chemistry ,31, pp. 455-461, 2010
[6] Irwin, J.J. & Shoichet B.K., "ZINC – A Free Database of Commercially Available Compounds for Virtual Screening," Journal of Chemical Information Modelling, vol. 45, pp. 177-182, 2005
[7] Wibisono, A., Suhartanto, H., Yanuar, A., "Cloud computing model and implementation of molecular dynamics simulation using Amber and Gromacs", Proc. of 2012 International Conference on Advanced Computer Science and Information System, pp. 31-36, 2012
[8] Hilman, M., Suhartanto, H., & Yanuar, "Performance Analysis of Embarassingly Parallel Application on Cluster Computer Environment: A Case Study of Virtual Screening with Autodock Vina 1.1 on Hastinapura Cluster". arXiv preprint arXiv:1305.3123, 2013
[9] H. Suhartanto, A. Yanuar, A. Wibisono,"Performance Analysis Cluster and GPU Computing Environment on Molecular Dynamic Simulation of BRV-1 and REM2 with GROMACS", International Journal of Computer Science Issues, vol. 8, no. 3, 2011
[10] Suhartanto, H., Yanuar, A., Hilman, M. H., Wibisono, A., & Dermawan, T. "Performance Analysis Cluster Computing Environments on Molecular Dynamic Simulation of RAD GTPase and LOX-Curcumin Molecules with AMBER". International Journal of Computer Science Issues (IJCSI), 9(2), 2012
[11] HPC Challenge Benchmark. [Online]. Available: http://icl.cs.utk.edu/hpcc/ accessed December 20, 2011.
[12] Demzar, J. "Statistical Comparisons of Classifiers over Multiple Data Sets, " Journal of Machine Learning Research, vol. 7, pp. 1-30, 2006