■ 1545

The Optimal High Performance Computing Infrastructure for Solving High Complexity Problem

Yuliant Sibaroni*[1], Fitriyani[2], Fhira Nhita[3]
[1,2,3]School of Computing, Telkom University, Bandung, Indonesia, (022) 7564108
*Corresponding author, e-mail: ysibaroni@gmail.com[1], fitriyani@telkomuniversity.ac.id[2],
fhiranhita@telkomuniversity.ac.id[3]

Abstract
The high complexity of the problems today requires increasingly powerful hardware performance. Corresponding economic laws, the more reliable the performance of the hardware, it will be comparable to the higher price. Associated with the high-performance computing (HPC) infrastructures, there are three hardware architecture that can be used, i.e. Computer Cluster, Graphical Processing Unit (GPU), and Super Computer. The goal of this research is to determine the most optimal of HPC infrastructure to solve high complexity problem. For this reason, we chose Travelling Salesman Problem (TSP) as a case study and Genetic Algorithm as a method to solve TSP. Travelling Salesman Problem is belonging often the case in real life and has a high computational complexity. While the Genetic Algorithm (GA) belongs a reliable algorithm to solve complex cases but has the disadvantage that the time complexity level is very high. In some research related to HPC infrastructure comparison, the performance of multi-core CPU single node for data computation has not been done. The current development trend leads to the development of PCs with higher specifications like this. Based on the experiments results, we conclude that the use of GA is very effective to solve TSP. the use of multi-core single-node in parallel for solving high complexity problems as far as this is still better than the two other infrastructure but slightly below compare to multi-core single-node serially, while GPU delivers the worst performance compared to others infrastructure. The utilization of a super computer PC for data computation is still quite promising considering the ease of implementation, while the GPU utilization for the purposes of data computing is profitable if we only utilize GPU to support CPU for data computing.

Keywords: TSP, Genetic Algorithm, cluster, GPU, HPC, multi-core CPU

1. Introduction

The requirement of High-Performance Computing (HPC) in various fields is increasingly urgent. In wheater, biology, aero, and nuclear reactor power systems modeling and simulation, support of high-performance computing is needed to get more quickly time and precisely results [1, 2]. Weather modeling system that uses weather data from the entire region in various parts of the world also requires very high computing support. For example, modelling in Coupled AGCMs at 50km atmosphere and 1 deg ocean have computation complexity in Terascale, while in Earth System Models at 10 km atmosphere and 1/10 deg ocean have computation complexity in Petascale and in most complexity problem, Full earth system models with carbon feedback cycle at 1km atmosphere and 1/100 deg ocean have computation complexity in Exascale [1].

The current infrastructure that can be used to support high-performance computing among others Computer Cluster, Graphical Processing Unit (GPU), and Super Computer PC. In a computer cluster, a set of computers with a uniform platform connected to a LAN network so that they can work together to perform parallel computing. GPU is usually used for heavy graphics processing, but the trend at this present time is the utilization of GPU in data computing because the advantages of GPU that have more than 2000 core therein [3]. A super computer PC is basically a computer that created by the PC manufacturer with a very high specification. At this time, a super computer PC have 192 processor core and delivering 96 GFLOPS core clock speed [4]. Comparison between these three infrastructures is very interesting. Supercomputer PC has advantages in terms of ease of use, but it has drawbacks in terms of high price and a limited number of processor cores. Computer clusters have flexibility in terms of the number of processor cores dynamically according to user needs, while the

drawback is the complexity of the cluster development process. The advantages of GPU is in its price compare than two others infrastructure, while the lack of GPU is more complicated in programming implementation and the level of GPU clock speed is lower than the CPU clock speed.

Comparison of computing time and other factors in some HPC infrastructure has been studied by several groups of researchers who aim to obtain the most optimal HPC infrastructure. Baker And Buyya [5] conclude that the use of computer clusters have advantages over the use of a dedicated parallel supercomputer that is associated with a lower price and incremental growth factor. Comparison between the CPU and GPU in cluster show that the performance of the GPU in Anomaly Detection in Hyperspectral Images is superior to the CPU clusters [6]. This is consistent with the utility of GPU that dedicated to image processing. Whereas in the case of cryptography, GPU processing time in data computing is superior (faster) than the single-core CPU processing time but worse than the performance of multi-core CPU to process the same data [7]. Performance multi-core CPU cluster (multi-core multi-node), is also more significantly superior than green-based cluster environment (cluster node with low power consumption: Raspberry Pi) for data computing [8]. In the cluster infrastructure, computation of single-core CPU is more effective than multi-core CPU for low complexity problem, while for high complexity problem, computation of multi-core CPUs is more effective than single-core CPU linearly with the number of nodes in the cluster [9].

By analyzing several studies related to the HPC infrastructure comparison, it can be concluded that a comparison of several HPC infrastructure undertaken aim to determine the most optimal infrastructure where the type of data that used (image / non-image) also affect the performance obtained and determine the appropriate infrastructure to be used. Overall, the programming that used in these studies is a parallel programming. In general, the performance of multi-core (multi-node) CPUs in a cluster infrastructure still has superior performance especially for computation with high complexity problems. But of all these studies, the evaluation of the performance of high complexity problems computation using multi-core CPU in single-node is not done. The performance of multi-core CPU in single-node is very interesting for further analysis considering the trend of the use of a PC or notebook with multi-core CPU specifications has also increased at present.

A super computer basically is also a PC with a multi-core CPU with very high specifications. Multi-core single-node PC just did serially where most users are already quite familiar with this serial programming. In this serial programming, the user does not need to perform parallel processes division manually but will be set automatically by the system.

This study aims to complement previous studies, especially related to the performance of multi-core single-node which represents a simple form of a super computer. Computation of high complexity data will be done using a multi-core CPU single-node serially and will be compared with single-core multi-nodes in parallel and multi-core GPU in parallel. Performance comparison of multi-core single-node serially infrastructure, multi-core parallel single-node and multi-core GPU in parallel infrastructure can also be considered as a comparison between three HPC infrastructure i.e. supercomputers, CPU cluster and GPU on a small scale. Comparison of the three infrastructures will be done in high complexity problem, where the Traveling Salesman Problem (TSP) route using genetic algorithms is taken in this study. The main contribution of this research is to give recommendation the optimal of the High-Performance Computing infrastructure to solve high computation problem.

2. Related Work
The research involved in this study includes many areas which associated with TSP, genetic algorithms, and HPC. The field of HPC is the most important concern in this study.

2.1. Research in Computer Cluster
Computer Cluster at the era of the 80s is a very fancy stuff for a college or laboratory and simply belongs for big companies only. Since the 2000s where the price of a PC has become quite affordable and high-performance computing needs increasingly needed, many laboratories then try to build their own computer cluster. Nowadays, there are many researches in high-performance computing field from among academia.

Research in the field of HPC using cluster computers was conducted by a number of researchers in the wide area. Barret, et al., examine simple and more complex matching MPI cores to perform operations [10]. In their research, Barret et al. analyzes the message rates delivered by current low-power general-purpose processors and compares them to the current high single-thread performance processors. Utilization of computer cluster is also applied in the networking field. Zounmevo et al. use MPI as a network transport for a large-scale [11]. They implemented MPI in three programming tool i.e. Open MPI, MPICH and MVAPICH. Handling complex data types are also examined by Traf in data type normalization [12].

2.2. Research in GPU
GPU utilization in computation today's has also been carried out by a number of researchers. Lefohn et al. develop a library for accessing data structures are generic and efficient GPU [13]. Mendez-Lojo, et al., utilizing the GPU to run irregular algorithms that operate on pointer-based data structures such as graphs [14]. The result is average speedup of 7x compared to a sequential CPU implementation and outperforms a parallel implementation of the same algorithm running on 16 CPU cores.

2.3. Research in Comparison between CPU, GPU, and Cluster
Comparison GPU and CPU clusters have been made by Abel Paz and Antonio Plaza 2010 [6] in the process of Anomaly Detection in Hyperspectral Images. In accordance with the utility of GPU that dedicated for graphics processing, the results obtained show that the performance of the GPU cluster in Anomaly Detection in Hyperspectral Images was superior to CPU cluster up to 32 nodes. CPU and GPU utilization in the cluster infrastructure also conducted by Mark, et al., [7] to large-scale computation in the case of cryptography. The experimental results show that the GPU processing time in data computing superior (faster) than a single-core CPU processing. While the best performance is obtained from the use of multi-core CPU to process the same data.

Each HPC infrastructure can be combined with another infrastructure in its implementation to obtain a more optimal result. Wang, et al., examine the level of efficient coordination mechanisms to handle parallel requests from multiple hosts of control to a GPU under hybrid programming [15]. In another study, Aji et al. shows that the utilization of the GPU-integrated MPI solutions, in epidemiology simulation can improve the performance up to 61.6% and can also improve the performance of a seismology modeling application up to 44%, when compared with traditional hybrid MPI+GPU implementations [16]. In another similar study, Choi et al. optimized CPU–GPU hybrid implementation and a GPU performance model for the kernel-independent Fast Multipole Method (FMM). In the best case, the achieve a speedup of 1.5× compared to GPU-only implementation [17].

2.4. Research in TSP
The Traveling Salesman Problem (TSP) is among the most famous NP-hard optimization problems. Research for TSP cases is generally much in terms of its mathematical aspects. Bartal et al. show that the algorithmic tractability of metric TSP depends on the dimensionality of the space and not on its specific geometry [18]. In another study, Fekete et al. can solve the Fermat-Weber-Problem (FWP) with high accuracy in order to find a good heuristic solution for the Maximum Weighted Matching Problem (MWMP) [19]. Bjorklund et al. show that the traveling salesman problem in bounded-degree graphs can be solved in time $O((2−\varepsilon)n)$, where $\varepsilon>0$ depends only on the degree bound but not on the number of cities, n [20].

3. Methods
The method that we used to solve TSP problem is Genetic Algorithm (GA). This genetic algorithm is implemented in three HPC infrastructure i.e. supercomputer, GPU, and computer clusters. We use three HPC infrastructure with similar price because the final goal of this research to obtain the most optimal HPC infrastructure economically. The parallel computing is implemented in GPU, and computer clusters while serial computing is implemented in supercomputer.

3.1. Hardware Infrastructure

In this research, we use a CPU with the specifications Intel core i5 @ 1.7 GHz, 4 cores and 4 GB of RAM to implement multi-core single-node serially and multi-core single-node parallel. While to implement multicore GPU, we use a GPU nvidia GTX 670 with 1344 cores and 980 MHz processor clock.

3.2. TSP

TSP is one issue that has a high complexity or NP-Complete. The idea of the TSP is to find the shortest route from the collection of the city, visiting the city exactly once and return to the city of origin. TSP is divided into two types, standard (symmetric) and asymmetric TSP. For example in symmetric TSP, given graph with vertex and certain node and inter-node weights, the result is a possible route permutation of the number of cities. When the number of very large cities, the complexity of time to find the cost of each route will be very large. Graph types are addressed in this study is a complete graph, that all the city (nodes) are connected with each other.

3.3. Genetic Algorithms

Genetic Algorithms is inspired by the theory of evolution and genetics. Solutions or models produced in GA form containing individual chromosomes or genes. The evolution of the GA process begins determining individual representation, then do the decoding for each chromosome. The next process is the initialization of the population. Each chromosome will be evaluated and selected based on the value of fitness as parents owned. A pair of parents will produce a child (new individual) of the crossover. The new individual will have mutations in some it's genes and configures a new nature that is really different from the genes of their parents. The resulting offspring will then be selected to replace the parental chromosomes in the process of forming the next generation.

3.3.1. Individual Representation and Crossover

In Genetic Algorithms (GA), an individual or chromosomes can represent as binary, real or integer. The representation that used in TSP is permutation representation, where the value of each gene is different from each other because each gene describes each city that has been visited. We use order crossover methods in this study. The purpose of this order crossover is to prevent the same city passed more than one time.

3.3.2. Fitness Function

Fitness function measures the degree of effectiveness of an individual as the solution of the system. Individuals with poor fitness value (small) will be eliminated in the next process. Individuals with good fitness values (large), likely to be a system solution. Fitness values used in this study can be seen in equation (1).

$$fitness\left(i\right)=\frac{1}{b+dist(i)}$$

(1)

Where

I	: individual i-th
B	: initial weight (small number)
dist (i)	: the amount of the distance between the cities i and i+1 on individual i, i = 1, ..,(the size of the city-1).

3.4. Parallel GA

There are several processes are carried out in GA like initialization population, individual evaluation, crossover, mutation, the formation of a new generation. In this study, individual evaluation process is conducted as a parallel process. This process is implemented in parallel on GPU and computer clusters.

4. Results and Analysis

The implementation of the genetic algorithm for TSP is conducted using three HPC infrastructure i.e. the GPU, supercomputers and computer clusters. Experiments performed with several parameters of GA. Types of GA's parameter settings can be seen in Table 1. TSP data that used in this study is obtained frorm 101-city problem by Christofides-Eilon.

Table 1. Parameters of Genetic Algorithm

Parameters	Values
Size of Population	100, 150
Prob. of Crossover	0.5, 0.7, 0.9
Prob. of mutation	0.1, 0.3, 0.5

4.1. Experiment Results Using Multi-Core GPU

By using the same genetic algorithm parameter settings such as Table 1, the results obtained in parallel genetic algorithm implementation using the GPU was shown in Figure 1. The experimental results showed that the time to solve TSP by using genetic algorithm and GPU infrastructure range from 0.5 seconds to 1.3 seconds.
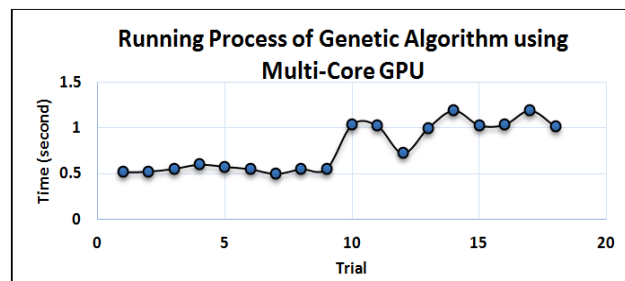

Figure 1. Running Process of Genetic Algorithm using Multi-Core GPU

4.2. Experiment Results Using Multi-Core Single-Node Serially

The result of GA implementation for TSP using GPU showed better compared to an implementation using a regular PC in our previous research. In this experiment, GA implementation for TSP is conducted using multi-core single node infrastructure serially. By using the configuration parameters such as the configuration parameters of the GPU, detailed results of the experiment can be seen in Figure 2. The experimental results using multi-core single node infrastructure serially shows that processing time that obtained range from 0.06 to 0.27 seconds.
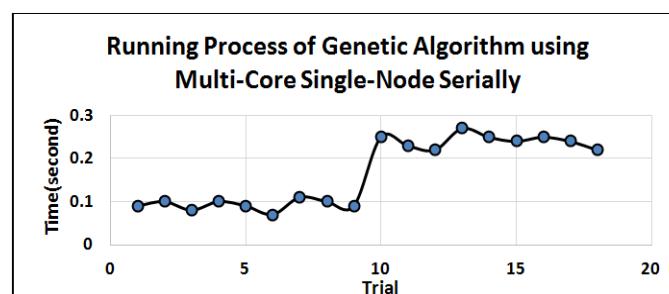

Figure 2. Running Process of Genetic Algorithm using Multi-Core-Single-Node Serially

4.3. Experiment Results Using Multi-Core Single-Node in Parallel

Last HPC infrastructure that used in our research is multi-core single-node in Parallel with MPI mechanism. Genetic Algorithm for TSP must be adjusted in parallel to be processed in this infrastructure. By using parameters as the same previous configuration, detailed results of the experiment can be seen in Figure 3. The experimental results for TSP-AG using multi-core single-node in Parallel showing the time around the value of the process gained 0.06 to 0.25 seconds, not far in contrast to multi-core single-node serially .
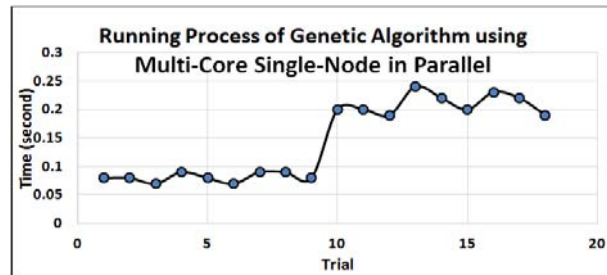

Figure 3. Running Process of Genetic Algorithm Using Multi-Core-Single-Node in Parallel

4.4. Comparison of 3 HPC infrastructure and others Research

To see a detailed comparison between the three HPC infrastructures, all results will be displayed in one graph, as shown in Figure 4. Referring to Figure 4, the best time results of the implementation TSP-GA is using multi-core single-node in parallel, although not significant compared to multi-core single-node serially. The worst processing time is obtained by the multi-core GPU.
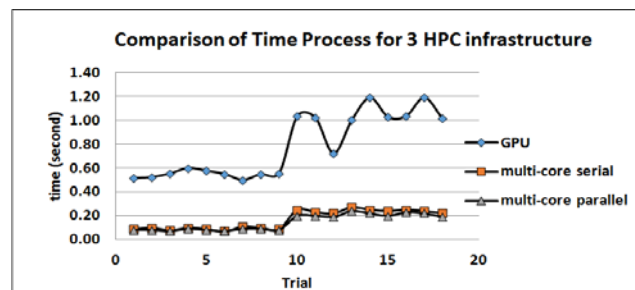

Figure 4. Running Process of Genetic Algorithm using 3 HPC infrastructure

Compared to the results obtained by other researchers, the experimental results obtained related to the advantages of multi-core CPU single node in parallel compared to multi-core GPU is still in line with the results obtained Mark, et al., [3] in which the best performance is obtained for multi-core CPU in the clusters infrastructure . Whereas for the results of multi-core CPU single-node serially certainly complement the results obtained Mark, et al., since similar experiments have not been done by Mark, et al. The experimental results related to the comparison between multi-core CPU single-node in parallel versus multi-core CPU single-node serially also similar to the results obtained by Rahman for high complexity problem computation.

Specific results that differentiate with other research infrastructure are related to the three comparisons made in this experiment. Although the performance of multi-core single node serially is not the best; but the use of multi-core single node serially still very worthy used for computation of data with high complexity. In general it can be concluded, the utilization of high specification PC (super computer PC) for data computation serially is still quite feasible and efficient than building clusters that are specifically used for computing data. Another advantage

is in programming aspect that quite done serially, which means the programming language used is also more varied.

5. Conclusion

It can be concluded that the use of GA is very effective to solve TSP compared to the brute force method ever used other research in solving TSP. By using similar HPC infrastructure specifications, the use of multi-core single-node in parallel for solving high complexity problems is better than the two others infrastructures. The processing time using multi-core single-node serially slightly below compare to multi-core single-node in parallel, while GPU delivers worst performance compared to others infrastructure. In general it can be concluded, utilization of a super computer PC for data computing is still quite promising considering the ease of implementation. While the GPU utilization for data computing is only promising when we use GPU to support data computing beside CPU but developing the special HPC infrastructure based on GPU is not profitable.

Acknowledgements

References

[1] P Nyberg. *HPC Trends and Directions in the Earth Sciences.* In ECMWF HPC workshop. 2014.
[2] PJ Turinsky. Advances In Multi-Physics And High Performance Computing In Support Of Nuclear Reactor Power Systems Modeling And Simulation. *Nucl. Eng. Technol.* 2012; 44(2): 103-112.
[3] GEFORCE GTX 1080. 2016.
[4] S Mcintosh-smith. A Next-Generation Many-Core Processor With Reliability, Fault Tolerance And Adaptive Power Management Features Optimized For Embedded And High Performance Computing Applications. 2016.
[5] M Baker, R Buyya. Cluster Computing: The Commodity Supercomputer. 1999; 29: 551-576.
[6] A Paz, A Plaza. Clusters versus GPUs for Parallel Target and Anomaly Detection in Hyperspectral Images. *EURASIP J. Adv. Signal Process.* 2010; 2010.
[7] M Marks, J Jantura, E Niewiadomska-szynkiewicz, P Strzelczyk, K Góźdź. Heterogeneous GPU & CPU Cluster For High Performance Computing. *Comput. Sci.* 2012; 13(2): 63-79.
[8] A Ashari, M Riasetiawan. High Performance Computing on Cluster and Multicore Architecture. *TELKOMNIKA Telecommunication Computing Electronics and Control.* 2015; 13(4): 1408-1413.
[9] A Rahman. High Performance Computing Clusters Design and Analysis Using Red Hat Enterprise Linux. *TELKOMNIKA Indonesian Journal Electrical Engineering.* 2015; 14(3): 534-542.
[10] BW Barrett, SD Hammond, R Brightwell, KS Hemmert. The Impact of Hybrid-Core Processors on MPI Message Rate. *EuroMPI/ASIA'13.* 2013: 67-71.
[11] JA Zounmevo, A Afsahi, R Ross. Using MPI in High-Performance Computing Services. *EuroMPI'13.* 2013: 43-48.
[12] JL Traf. Optimal MPI Datatype Normalization for Vector and Index-block Types. *EuroMPI/ASIA'14.* 2014: 33-38.
[13] AE Lefohn, S Sengupta, JOE Kniss, R Strzodka, JD Owens. Glift: Generic, Efficient, Random-Access GPU Data Structures. *ACM Trans. Graph.* 2006; 25(1): 60-99.
[14] M Mendez-Lojo, M Burtscher, K Pingali. *A GPU Implementation of Inclusion-based Points-to Analysis.* PPoPP'12. 2012: 107-116.
[15] L Wang, M Huang, T El-Ghazawi. *Towards Efficient GPU Sharing on Multicore Processors.* PMBS'11. 2011: 23-24.
[16] AM Aji, LS Panwar, F Ji, M Chabbi, K Murthy. *On the Efficacy of GPU-Integrated MPI for Scientific Applications.* HPDC'13. 2013: 191-202.
[17] J Choi, A Chandramowlishwaran, K Madduri, R Vuduc. *A CPU – GPU Hybrid Implementation and Model-Driven Scheduling of the Fast Multipole Method.* In ACM.org. 2014.
[18] Y Bartal, LA. Gottlieb, R Krauthgamer. *The Traveling Salesman Problem : Low-dimensionality.* STOC'12. 2012: 663-672.
[19] P Fekete, H Meijer, I Science, D Mathematik, W Tietze. *Solving a ' Hard ' Problem to Approximate an ' Easy ' One : Heuristics for Maximum Matchings and Maximum Traveling Salesman Problems.* ACM. 2001; 1(212).
[20] ABJ Bjorklund, T Husfeldt, P Kaski, M Koivisto. The Traveling Salesman Problem in Bounded Degree Graphs. *ACM Trans. Algorithms.* 2012; 8(2): 1-13.