UNIVERSITI TEKNOLOGI MARA

IMPROVING PARALLEL SELF-ORGANIZING MAP USING HETEROGENEOUS UNIFORM MEMORY ACCESS

MUHAMMAD FIRDAUS BIN MUSTAPHA

Thesis submitted in fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science)

Faculty of Computer and Mathematical Sciences

March 2018

AUTHOR'S DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Teknologi MARA. It is original and is the results of my own work, unless otherwise indicated or acknowledged as referenced work. This thesis has not been submitted to any other academic institution or non-academic institution for any degree or qualification.

I, hereby, acknowledge that I have been supplied with the Academic Rules and Regulations for Post Graduate, Universiti Teknologi MARA, regulating the conduct of my study and research.

Name of Student	:	Muhammad Firdaus bin Mustapha
Student I.D. No.	:	2014490362
Programme	:	Doctor of Philosophy (Computer Science) – CS950
Faculty	:	Computer and Mathematical Sciences
Thesis Title	:	Improving Parallel Self-Organizing Map Using Heterogeneous Uniform Memory Access
Signature of Student	:	……,……
Date	:	March 2018

ABSTRACT

Self-organizing Map (SOM) is a very popular algorithm that has been used as clustering algorithm and data exploration. SOM consists of complex calculations where the calculation of complexity depending on the circumstances. Many researchers have managed to improve online SOM processing speed using Heterogeneous Computing (HC). HC is a combination of Central Processing Unit (CPU) and Graphic Processing Unit (GPU) that work closely together. Standard HC can be represented by CPU and GPU accessing separate memory blocks. In spite of excellent performance using standard HC, there is a situation that causes computer hardware underutilized when executing online SOM variant. In details, the situation occurs when number of cores is larger than the number of neurons on map. Moreover, the complexities of SOM steps also increase the usage of high memory capacity which leads to high rate memory transfer. This situation is caused by the standard HC implements "deep copies" in storing processing objects which lead to communication latency. Recently, combination CPU and GPU that integrated together on a single chip are rapidly attractive the design paradigm for recent platform because of their remarkable parallel processing abilities. This kind of microprocessor is based on Heterogeneous Unified Memory Access (HUMA) model. This model allows both CPU and GPU to access and store into the same memory location which avoids redundant copies of objects by "deep copies" method. Therefore, the main goal of this research is to reduce computation time of SOM training through implementing on HUMA platform and improve GPU cores utilization. This research has three main objectives to be achieved. Firstly, this research attempts to study the processing natures of original SOM algorithm on standard HC platform. Secondly is to model an enhanced parallel SOM on HUMA-GPU platform and adapting multiple stimuli approach in order to improve the processing speed. Lastly is to evaluate the enhanced parallel SOM in terms of performance accuracy, efficiency, and scalability. This research attempts to improve the processing of SOM algorithm through three stages. The research works start with conducting a preliminary study on sequential SOM algorithm. The research continues to design a parallel SOM architecture based on literature study and implements on two types of architecture; standard HC and HUMA model. Finally, this research designs and implements an enhanced parallel SOM architecture through combining two parallel methods which are network and data partitioning. The combination of the two methods are realized via adapting multiple stimuli approach. This research employs datasets that are acquired from UCI repository. As a result, the enhanced parallel SOM that executed on HUMA platform is able to score up to 1.27 of speed up overall for large map size compared to standard parallel SOM. The proposed work also scores better for smaller map size with scored up to 1.03 of speed up overall compared to standard SOM on the identical platform. Accordingly, the proposed work is able to offer a better solution for small to medium sized of data analysis software. Overall, the solution is enhanced through utilizing recent hardware technology and improved method.

TABLE OF CONTENTS

CHAPTER ONE
INTRODUCTION

1.1 Research Background

Self-organizing Map (SOM) is an unsupervised neural network that has been used as data analysis method. It is being widely used and applied to solve clustering and data exploration problems in various domain areas, with remarkable abilities to remove noise, outliers, and deal with missing values (Teuvo Kohonen, 2013). There were many researches have been found in the literature that used SOM to solve clustering problem (Claveria, Monte, & Torra, 2016; Llanos et al., 2017; Matic et al., 2017; Tsai, Huang, Cheng, Shao, & Chang, 2016). Despite its excellent performance, there are problems related to slow processing when visualizing large map size (McConnell, Sturgeon, Henry, Mayne, & Hurley, 2012). This imposed heavy workload on the processor especially when dealing with winner-search and updating weightage of neurons on the map (Teuvo Kohonen, 2013). On the other hand, the datasets dimension also have high influence in SOM processing (Hasan et al., 2014).

This situation attracts much interest among researchers to improve SOM processing by parallelizing the algorithm. Among the common ways to parallelize SOM are network or map partitioning (Khalifa, Girau, Alexandre, & Bedoui, 2004; Kurdthongmee, 2008, 2015) and data or example partitioning (Abe, Hamano, Kanaya, Wada, & Ikemura, 2009; Moraes, Botelho, Filho, & Gaya, 2012; Sul & Tovchigrechko, 2011). However, there also efforts to parallelize SOM algorithm through combining both network and data partitioning (Mojarab, Memarian, Zare, Hossein Morshedy, & Hossein Pishahang, 2014; Richardson & Winer, 2015) with the interest to gain advantages of both parallelism. In the meantime, most of research works on improving SOM are aimed to achieve efficiency and scalability in their proposed works. In details, proposed parallel SOM that efficient should be faster in term of processing than the previous version (Dzemyda & Kurasova, 2004; Garcia, Prieto, & Pascual-Montano, 2005; MacLean & Valova, 2007). Meanwhile, some research works are attempting to increase the utilization of processing elements in executing the SOM algorithm (Dlugosz, Kolasa, Pedrycz, & Szulc, 2011;