UNIVERSITI TEKNOLOGI MARA

ANALYSIS OF EVOLUTIONARY COMPUTING PERFORMANCE VIA MAPREDUCE PARALLEL PROCESSING ARCHITECTURE

AHMAD FIRDAUS BIN AHMAD FADZIL

Thesis submitted in fulfillment
of the requirements for the degree of
Master of Science

Faculty of Computer and Mathematical Sciences

October 2014

ABSTRACT

Evolutionary computation (EC) is a method that is ubiquitously used to solve complex computation. Examples of EC such as Genetic Algorithm (GA) and PSO (Particle Swarm Optimization) are prevalent due to their efficiency and effectiveness. Despite these advantages, EC suffers from long execution time due to its parallel nature. Therefore, this research explores the prospect of speeding up the EC algorithms specifically GA and PSO via MapReduce (MR) parallel processing framework. MR is an emerging parallel processing framework that hides the complex parallelization processes by employing the functional abstraction of “map and reduce”. The performance of the parallelized GA via MR and PSO via MR are evaluated using an analogous case study to find out the speedup and efficiency in order to measure the scalability of both proposed algorithms. Comparisons between GA via MR and PSO via MR are also established in order to find which EC algorithm scales better via MR parallel processing framework. From the results and analysis obtained from this research, it is established that both GA and PSO can be efficiently parallelized and shows good scalability via MR parallel processing framework. The performance comparison between GA via MR and PSO via MR also shows that both algorithms are comparable in terms of speedup and efficiency.

ACKNOWLEDGEMENTS

In the name of Allah, the Most Gracious, Most Merciful.
Alhamdulillah, praises to Allah whom with His blessings allowing me to complete this thesis.

I would like to express the sincere gratitude to Dr. Noor Elaiza Abd Khalid for providing me with boundless dedication in supervising my research throughout the years. Without her constant support and guidance, this thesis will never bear its fruit and produce the envisioned outcome. Special thanks credited to my co-supervisor Assoc. Prof. Dr. Mazani Manaf in which his critics and support keep me stay on track with my research.

Special mention also to Pn Noorhayati and Pn Siti Arpah for allowing me to use their existing parallel computing resources which in turn sanctions me to build the parallel architecture essential to this research.

Deepest appreciations to my parent Ahmad Fadzil Mahmud and Sofiah Hashim for the unparalleled support which never failed to afford me with continuing and unwavering motivations. Thank you also to the rest of my family, friends and colleagues for their understanding, encouragement, and blessing that accompanied me during the course of my studies.

Last but not least, a sincere thanks to Universiti Teknologi MARA (UiTM) and Ministry of Higher Education (KPT) for providing me with the Tenaga Pengajar Muda Scholarship which has helped me greatly during the period of my studies.

TABLE OF CONTENTS

CHAPTER ONE
RESEARCH BACKGROUND

1.1 RESEARCH OVERVIEW

Evolutionary computing (EC) is a field of computer science that adapts the theory of Darwinian evolution to optimize computing problems (Eiben & Smith, 2003). The basis of EC revolves around the process of natural evolution such as random variation, competition, and reproduction (De Jong et al., 1997). The process of EC is widely applicable for various types of problems due to the fact that evolution itself is an optimization process (De Jong et al., 1997). Two of its most prevalent algorithms, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) have been used ubiquitously to optimize problems in numerous fields of study (Eberhart & Shi, 1998; Eiben & Smith, 2003; Settles & Soule, 2005; Premalatha & Natarajan, 2009). Despite the robustness of EC, expensive computing resources are required due to its parallel nature (Rajan & Nguyen, 2004; Massa et al., 2005; Zhu & Wang, 2010; Farmahini-Farahani et al., 2010). This indicates that EC would perform better in parallel where it can be executed simultaneously compare to that of a normal serial execution. Therefore, EC has since become a popular subject of parallelization using various parallel processing techniques and architectures (Cantú-Paz, 1998).

Parallel processing allows access to more computing resources compare to the traditional serial approach of computing. Examples of parallel processing architectures include high-end server and mainframe that are very expensive and only applicable to large company and corporations. Nevertheless, parallel processing can also be employed in a cost-effective manner. This is achieved by combining collection of commodity off-the-shelf (COTS) computing nodes to emulate a high-end-server or mainframe (Brightwell et al., 2000). Parallel processing has gained a huge ground with the emergence of various frameworks such as Message Passing Interface (MPI) (Dongarra et al., 1993) and OpenMP (Dagum & Menon, 1998). However, the frameworks are known for their complex parallelization process specifically in terms of managing and distributing the resources