Proceedings of the 2009 IEEE 9th Malaysia International Conference on Communications
15 -17 December 2009 Kuala Lumpur Malaysia

Optimizing LZW Text Compression Algorithm via Multithreading Programming

Ling Sun Tan[1], Sei Ping Lau[2], Chong Eng Tan[3]
Faculty of Computer Science and Information Technology,
University Malaysia Sarawak,
94300 Kota Samarahan, Sarawak, Malaysia
daphne.lingsun@gmail.com[1], splau@fit.unimas.my[2], cetan@fit.unimas.my[3]

***ABSTRACT* - Due to the emerging multimedia technology, multimedia files such as text, image, audio and video files are widely used. These multimedia files take hundreds time more space as compared to early day's media files. Thus, demand for efficient compression algorithm is in great needs. Currently, a lot of general purpose multi-core processor machines and systems are widely available. However, many compression algorithms have not been taking advantage of being optimized for these processors. This paper explores the multithreaded compression algorithm to take advantages offered by multi-core processor.**

Keywords – text compression, parallel, multithreaded programming

I. Introduction

Compression or zip algorithms have become more and more important due to the widely use of multimedia technology. Multimedia files such as text, image, audio, and video demand much more storage space due to the growing of the multimedia files' size, which may easily require hundreds times more space as compared to early day's media files. Thus, a compression program is essential to make the file size smaller and directly reduce the consumption of expensive resources, such as hard disk space or transmission bandwidth. However, the performance of compression program can be further improved by incorporating multithreaded programming which takes advantage of the new multi-core processor architecture.

Today, a lot of general purpose desktop computer has been equipped with multi-cores processor [2] such as the Intel's Core Processor family and the AMD Athlon X2 Dual-Core processors. With multi-core technology, tremendous processing speed is offered but it does not really improve the data compression algorithm performance if the compression program is based on sequential programming design. Undeniably, many current compression programs have not been optimized for this technology and it is a waste of processor throughput if there was only one thread being created to manage all incoming tasks on its own. Multithreading programming model is a new turning point to exploit the high throughput of the new multi-core processor architecture. Therefore, a compression program that can take full advantage of multithreaded computing should be designed to further optimize the overall processing performance.

This paper studies and enhances the Lempel-Ziv-Welch (LZW) algorithm via multithreading programming especially on the performance gained through different number of threads. On our Dual Quad-Core Xeon processor system, the preliminary result shows 87% performance improvement while 8 compression threads are executed simultaneously. Various sizes of datasets were used and the average time of 10 compression and decompression processes are used for result comparison.

The rest of this paper is organized as follow. Section 2 summarizes related works by other researchers. Section 3 provides an overview of LZW algorithm. Section 4 describes the **M**ulti**T**hreaded **LZW** (MTLZW) compression and decompression implementation. Section 5 discusses result analysis and comparison. Section 6 concludes the works of this paper. Section 7 suggests possible future works.

II. Related Works

A number of previous works on parallel data compression are presented in [4, 5, 6, 7]. In these research works, multiprocessors platform is assumed for implementing the parallel version of data compression algorithm. In [4] and [6], the authors parallelized the string parsing process together with static dictionary which come with prefix property. In [4], the *k-optimal* parsing was proposed as compared to the initial *optimal* parsing which is presented in [6]. The parallel version of LZSS and LZW coding are proposed and presented in [5]. The VLSI with systolic arrays has been used in [7] to implement the parallel algorithm by using textual substitution technique for both static and dynamic dictionary scheme.

978-1-4244-5532-4/09/$26.00 ©2009 IEEE