ISSN:0975-9646

Darshan R et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 2865-2867

Acceleration of LSB Algorithm in GPU

Darshan R[1], Prabu R[2], Divya M[3]

[3]*Assistant Professor,Department of Information Technology,*
Amrita Vishwa Vidyapeetham,
Coimbatore, Tamil Nadu, India.

***Abstract-* This paper presents a method for acceleration of LSB (Least Significant Bit) Algorithm in GPU (Graphics Processing Unit) using a programming model called CUDA. CUDA is a state-of-the-art parallel computing architecture developed by nVIDIA. CUDA allows the programmers to access the GPU directly by invoking the Kernel. In Image Steganography, parallelization of computations to a single pixel and the hybrid mix of message passing and shared memory access routines allows us to accelerate the LSB algorithm and thereby reducing the runtime of the program.**

***Keywords-* CUDA, GPU, LSB Algorithm, Image Steganography, Parallel Processing Architecture, Acceleration of Algorithms, Parallel Computing.**

I. INTRODUCTION

Steganography is a technique which deals with encoding information in an image without making any significant visible changes to it. The goal is to hide an information within the cover image, such that the encoded file's existence is concealed. It is of immense use because it allows the sender and the receiver to protect the identity of the file. Several methods have been proposed for image based steganography, LSB being the most efficient and simplest of all. Nowadays, plagiarism is a big threat in photography and identity spoofing is prevalent in many places. To curb this, the solution used is Steganography.

CUDA is a general purpose parallel computing architecture that leverages the parallel compute engine in nVIDIA Graphics Processing Units (GPUs) to solve many complex computational problems in a fraction of the time required on a CPU. It includes the CUDA Instruction Set Architecture (ISA) and the parallel compute engine in the GPU. This is the platform used to implement the LSB Algorithm. Parallelization of LSB Algorithm can immensely benefit the real-time applications using steganographic techniques.

II. TOOLS AND REQUIREMENTS

To implement the LSB Algorithm in CUDA the following tools are necessary,

A. *Software Requirements*

- nVIDIA CUDA 5.5 Toolkit
- Microsoft Visual Studio 2012 (for Windows 7, 8, 8.1) or Eclipse (for Linux)

B. *Hardware Requirements*

- A CUDA capable nVIDIA Graphics Card
- Preferably 2 GB DDR3 RAM or Greater
- Preferably Intel Core i3-2330M CPU @ 2.20 GHz processor or Greater

III. IMPLEMENTATION

Implementation of the LSB Algorithm can be done with inputs of an image of size M*N, where M represents number of pixels on the X-Axis and N represents number of pixels in the Y-Axis and an audio file which is chosen as .information file. The input cover image needs to be converted into a bitmap format, where each pixel is represent by the values taken by the RGB components. The audio file is read in binary format and stored in the LSB of the RGB components for the considered bitmap cover image in a cyclic order. Each pixel can store 3 bits of data from the considered audio file, where all the 3 bits are equally shared by RGB components in their LSB. Therefore I byte of the information is stored in 3 pixels of the cover image. The algorithm is implemented using CUDA (CUDA uses nvcc compiler which has C compiler with additional functions and keywords provided by nVIDIA)

IV. PARALLELIZATION

A. *Parallelization requirements and CUDA Architecture Support*

To parallelize the proposed algorithm we require M*N threads. Since these threads can execute independent of each other, we can parallelize LSB to maximum extent (.i.e.to a pixel level). These requirements can be satisfied by using nVIDIA Graphics Processing Unit, where there are numerous kernels capable of executing independent of each other. In the GPU there are several Blocks which contains 1024 grids and each grid carries 1024 threads or kernels. Therefore we have an ideal architecture to implement the parallelization of LSB Algorithm.

B. *Data Structures*

The byte can be stored in unsigned character, where the values range from 0 to 255. Collection of all bytes are stored in unsigned character array. The byte values are converted to Boolean when the LSB need to be altered to store the audio bits and they are again converted back to bytes and these bytes ate written to the file. To make GPU operational we need to allocate memory in the device (device represents GPU) and pass the unsigned chat array pointer to the __global__ function (__global__ is a keyword which is understood by the device).

C. Workflow in CUDA

The memory allocation for the device is followed coping the data in by global function call with M number of grids and N number of kernels. When the globalfunction is invoked the control is shifted from CPU to GPU. The LSB Algorithm is embedded in the global function with appropriate grid id and thread id. Once all the kernel finishes its execution, the control is transferred back to CPU.

V. PERFORMANCE ANALYSIS

The performance measure used here is the Speed-Up factor. Speed-Up can be defined as a ratio of Time Taken for the Execution of a set of instructions in Single Core is to Time Taken for the Execution of the same set of instructions in Multicore. For the calculation, five sample cover images are chosen. The information to be hidden are the five sample audio files corresponding to the respective cover images. LSB Algorithm is executed separately on both CPU AND GPU. The time taken for their execution is measured and analysed respectively. The information regarding the sample inputs are mentioned in TABLE I. The result we obtained is shown in TABLE II. The comparison of their execution time between CPU and GPU can be understood from FIGURE I.

TABLE I

SAMLE INPUTS TO LSB ALGORITHM

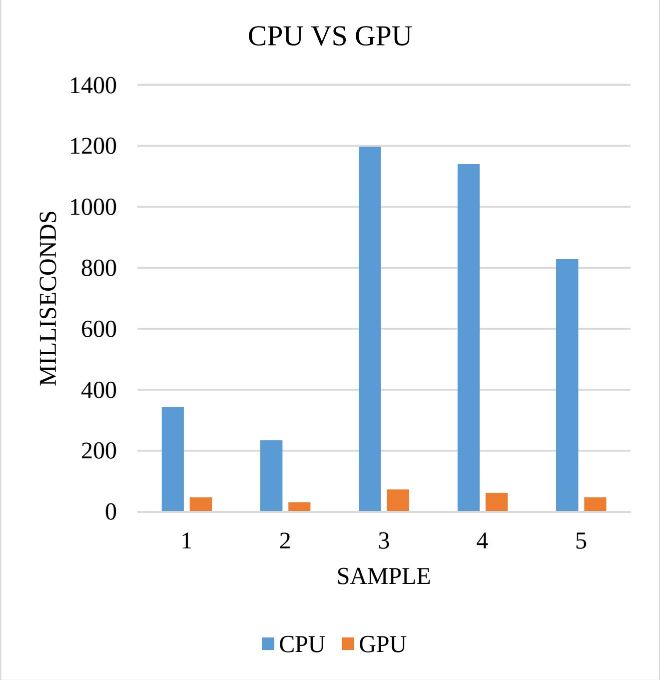
SAMPLE	COVER IMAGE (.BMP)	MESSAGE FILE (.MP3)	SIZE OF THE COVER IMAGE (IN KB)	SIZE OF THE MESSAGE FILE (IN KB)
1	Elephant	Elephant	481	10
2	Penguin	Penguin	480	8
3	Bike	Bike	482	36
4	River	River	481	32
5	Tiger	Tiger	480	23

TABLE III
RESULTS OF LSB ALGORITHM IN CPU AND GPU FOR THE SAMPLE INPUTS

SAMPLE	TIME TAKEN IN CPU (IN MS)	TIME TAKEN IN GPU (IN MS)	SPEED-UP
1	344	47	9.58
2	234	31	9.13
3	1197	73	41.96
4	1140	62	30.87
5	828	47	21.41

FIGURE I

COMPARISON OF TIME TAKEN FOR EXECUTION IN CPU AND GPU



VI. CONCLUSION

From the above statistics, we can observe that the execution time taken in GPU are just the fractions of the time when compared to the time taken for execution in CPU and thereby accelerating the LSB Algorithm up 40 times faster (on average 20 times faster). The true power of CUDA can only be realized in real-time application where numerous computations need to be done within the time bound. With the development of new and powerful parallel computing architectures, the real-time application developers are encouraged to implement their ideas in the parallel architecture and reap the benefits. Parallel computing is a strong tool and once they are used appropriately the benefits we achieve is beyond measures because we can save the most valuable and non-renewable resource, the time. Therefore we can further enforce the need for parallelization.

ACKNOWLEDGEMENT

We take this opportunity to express our profound gratitude and deep regards to our project guide Miss Divya M for her exemplary guidance, monitoring and constant encouragement throughout the course of this project. We also take this opportunity to express a deep sense of gratitude to Miss Sikha O K, for her cordial support, valuable information and marvellous guidance. We are obliged to staff members of Security and Networking Panel, for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of our project.

REFERENCES

[1] Patel, B., Samir, Pradhan, N., Shrikant, Ambegaokar, U., Saumitra, 'A Novel Approach for Implementing Steganography with Computing Power obtained by combining CUDA and MATLAB', International Journal of Computer Science and Information Security (IJCSIS), Vol. 6, No. 2, pp.133-137, 2009.

[2] Adnan Abdul-Aziz Gutub, 'Pixel Indicator Technique for RGB Image Steganography', King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia.

[3] Bloisi, Domenico, Ioochi, Luca, 'Image based Steganography and Cryptography', Sapienza University of Rome, Italy.

[4] Gupta, Shilpa, Gujral, Geeta, Aggarwal, Neha, 'Enhanced Least Significant Bit algorithm for Image Steganography', IJCEM International Journal of Computational Engineering & Management, Vol. 15, Issue 4, pp. 40-42, July 2012.

[5] Sorel, Y., 'Massively Parallel Computing Systems with Real Time Constraints, The Algorithm Architecture Ad equation Methodology', INRIA, Domaine de Voluceau, Rocquencourt B.P.105 78153 LE CHESNAY CEDEX FRANCE.

[6] Mohammad Tanvir Parvez, Adnan Abdul-Aziz Gutub, 'RGB Intensity Based Variable-Bits Image Steganography', IEEE Asia-Pacific Services Computing Conference, pp. 1322-1327, 2008.

[7] Yu-Chee Tseng, Member, IEEE, Yu-Yuan Chen, Hsiang-Kuang Pan, 'A Secure Data Hiding Scheme for Binary Images', IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 50, NO. 8, pp. 1227-1231, AUGUST 2002.

[8] Channalli, Shashikala, Jadhav, Ajay, 'Steganography, An Art of Hiding Data', International Journal on Computer Science and Engineering Vol.1 (3), 2009, pp. 137-141

.