International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-6, Issue-4, July-2018
DOI: 10.21276/ijircst.2018.6.4.1

GPU Implementation of Sales Forecasting with Linear Regression

Ayomide Yusuf, Shadi Alawneh

***Abstract*—Forecasting of sales is very important in any business as it helps managers to learn from historical data and make informed decisions. This generally involves intensive processes using spreadsheets which require inputs from all levels in an organization. This approach introduces bias and is generally not accurate. There are several methods that have been used in the past to forecast sales, such as Exponential Smoothing, Moving Average, and Autoregressive Moving Average (ARMA). Due to the nature of the data, it usually takes more time for these methods to analyze the sales data and make predictions. In this paper, the sales data is analyzed and predictions are made by using linear regression as implemented on the GPU to make the process faster. Sales forecasting is made possible by finding best fit line by linear regression techniques (i.e. linear convolution). To illustrate this process, simulated sales data was used. The sales forecasting with linear regression implementation using GPU was compared to the CPU implementation and a speedup of up to 7.557x was achieved.**

***Index Terms*—GPU, CUDA, Linear Regression, Sales Forecasting.**

I. INTRODUCTION

Executives today consider some kind of forecast in every decision they make. Sound predictions of requirement and style are no thirstier luxury items, but a necessity, if managers are to cope with seasonality, sudden changes in demand story and price-cutting tactical maneuver of the contest, strikes, and large cut of the economy [1]. This process is important as it will determine in the long run if the business will make profit or loss. Sale forecasting will allow the manager to know the right product to stock at the right time to avoid waste when some products expire. Forecasting will clearly show the managers the products that sell quickly in a particular time, it will help them to know the optimum price for a particular product to make profit and still make large sales in limited time. Sales forecast is the backbone of your business plan. Business growth is measured by its sales and forecast set standards for expenses, profits and growths. In this paper, sales forecasting was designed and implemented on the GPU.

A. Novel Contributions

To the best of our knowledge, our literature review of this subject shows there is no published work that used GPU to implement sales forecasting algorithms with linear regression. Therefore, this paper introduces a new way that GPU can be used to forecast sales using linear regression. With this newapproach, the process of forecasting is faster compared to the existing implementations which are not implemented in parallel.

B. Paper Outline

The rest of this paper is organized as follows- Section II discusses related research available, Section III explains some sales forecasting methods, issues, and limitations, Section IV describes GPU Programming Models, Section V introduces CUDA, Section VI describes the implementation process in this paper, Section VII presents the results and performance, while Section VIII presents the conclusion and future work.

II. RELATED WORK

There are several researchers who have to discuss sales forecasting and its application to real-life problems. Samaneh et al. [12] did a survey on retail sales forecasting and forecasting in fashion markets. They concluded that conventional forecasting methods face challenges in producing accurate sales data for new products and consumer-oriented goods. Henrik et al. [13] empirically explored and analyzed the attitudes towards sales forecasting management and the familiarity with forecasting techniques within a certain organization. A sales forecasting model of an automobile company using a Fuzzy BPN algorithm was successfully implemented by Rashmi et al. [14].

The goal of this work is to use linear regression to implement a faster sales forecasting process using GPU.

III. SALES FORECASTING METHODS AND LIMITATIONS

An increased number of companies are faced with challenges in connection with their sales forecasting. Among the most common examples are low forecasting accuracy, high complexity, lack of competencies in forecast, and heavy works process [2]. Sales forecasts will

Manuscript received June 18, 2018.

Ayomide Yusuf, Electrical and Computer Engineering, Oakland University, Rochester, MI, USA. E-mail: ayomideyusuf @oakland.edu

Shadi Alawneh, Assistant Professor, Electrical and Computer Engineering, Oakland University, Rochester, MI, USA. Phone No.:+12483702242, E-mail: shadialawneh@oakland.edu

not accurately predict the future because parameters that causes change in sales change rapidly. Despite this, prediction is needed in any business as it gives estimate of what to expect and the accuracy improves with time. Sales forecasting is also a tedious task as it involves volumes of data to process. In spite of this, it is a needed process in any business.

There are many sales forecasting methods that have been explored by financial and statistical researchers, listed below are some of the popular methods:

A. Jury of Executive Opinion

This method of sales agreement prediction is the oldest. One or more of the administrator who are experienced and have good cognition of the market factors predict the expected sales [3]. The executives will use experience and estimates to forecast sales figures. All factors, both internal and external that the executives have learnt over the years in the business are taken into consideration.

This method is simpleton, as experience and judgement are pooled together in taking a gross sales event prognosis figure. If there are many executives, their estimates are averaged in drawing the sales forecast

Advantages:

a) This method is simple and quick.
b) It does not require detailed data, just the experience of the executives.
c) There is economy.

Disadvantages:

a) It is not based on factual data.
b) This method is guess work and can lead to wrong forecasts.
c) It is difficult to draw a final conclusion from suggestions made by all executives.

B. Forecast Stages (Salesman Opinion)

This approach relies on the perceptiveness and suspicion of the gross revenue reps rather than on a mountain moving through pre-determined point. With forecast stages, reps make a personal projection about the outcome of any given sales agreement opportunity.

Since sales reps are the ones who deal directly with the customer, they could tell if a customer will buy a particular product at a particular time. Customers might even make promise to buy some products at a particular time if the reps always have this kind of information. The component here is that a rep makes judgement on what volume of sales to expect given certain conditions based on their experience. When this information comes at the beginning of a deal's lifespan or goods procurement, it can help managers and executive to get a long-range view of results. The sooner they have that information, the better their financial predictions will be.

Advantages:

a) The forecast is mostly accurate if the sales reps are truthful.
b) It requires fewer resources.

Disadvantages:

a) It is dependent on the honesty of the sales reps.
b) It cannot be quantified.

C. Forecast Stages (Salesman Opinion)

This method involves introducing products in an express geographical area and the issue is analyzed. Taking this result as a base, sales forecast are made. This examination is conducted as a sample on pre-test basis in Holy Order to understand the market response [3]. It involves taking data from a subset in the universal set and using it to predict the behavior of the whole set.

Advantages:

a) The forecast is based on actual results so it is reliable.
b) It can be used to introduce new products in a new location and test how people will react.
c) Management can understand the defects and take steps to rectify it.

Disadvantages:

a) It is a costly process.
b) It is time consuming.

D. Consumer's Buying Plan

Consumer, as a source of entropy, is approached to know their likely purchases during the period under a given set of Synonyms/Hypernyms (Ordered by Estimated Frequency) of noun condition. This method is suitable when there are few clients. This type of forecasting is generally adopted for industrial good. It is suitable for industry, which produce costly goods to a limited number of emptor s- middleman, retailers, potential drop consumer etc. A resume is conducted on a face to face basis or view method. This is because changes are constant while buyer behavior and buying conclusion change frequently [3]. This method predicts sales based on market surveys.

Advantages:

a) The information is directly from customers.
b) User's intention can be studied.

Disadvantages:

a) It is difficult to identify actual buyers.
b) It is costly.
c) Buyers may change their mind in future.
d) Customer's expectation cannot be measured exactly.

E. Market Factor Analysis (Linear Regression)

Linear regression is a linear approach for modeling the relationship between a scalar dependent variable Y and one or more explanatory variables (or independent variables) denoted as X [5]. It helps to estimate or predict the unknown values of one variable from the known values of another variable. It is called simple linear regression when there is one explanatory variable. In simple linear regression, we predict scores on one variable from the scores on a second variable. The variable we are predicting is called the criterion variable and is referred to as Y. The variable we are basing our predictions on is called the predictor variable and is referred to as X [6].

Linear regression consists of finding the best-fitting straight line through the points as shown in Fig. 1. The best-fitting line is called a regression line.

ISSN: 2347-5552, Volume-6, Issue-4, July-2018
DOI: 10.21276/ijircst.2018.6.4.1

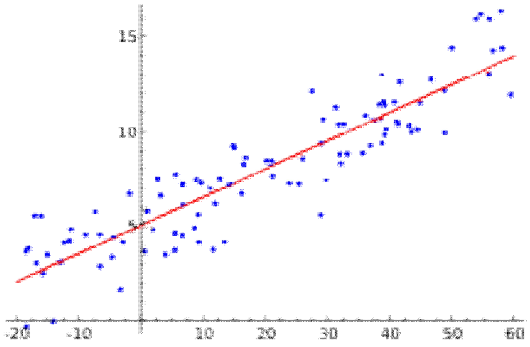


Fig. 1. Linear regression
(https://en.wikipedia.org/wiki/Linear_regression)

The formula for a regression line is:

$$Y' = bX + A$$

Where Y' is the predicted value, X is the given value, b is the slope of the line, and A is the Y intercept.

Advantages:
a) It is the most accurate method.
b) It is a sound method.
c) Market factors are analyzed in detail.

Disadvantages:
a) It is usually costly and consumes time.

IV. GPU PROGRAMMING MODELS

GPUs provide high computation power at low costs and have been described as desktop supercomputers [7]. The GPUs have been used for many general purpose computations due to their low cost, high computing power, and high availability [8]. It is widely used in applications that are computing intensive like Financial Analysis, Digital Audio Processing, Computer Vision, Medical Engineering etc.
CUDA from NVIDIA presents a heterogeneous programming model where the parallel hardware can be used in conjunction with the CPU [9]. It can be used as Bulk Synchronous Parallel (BSP) hardware with the CPU deciding the barrier for synchronization. There are three ways to develop software on GPU:

A. *Use of Libraries*

This involves the use of GPU libraries without in-depth knowledge of the implementation. These accelerated libraries follow standard API development, thus enable acceleration with minimal code changes. It usually offer high-quality implementations of functions encountered in a broad range of application. Examples of these libraries include cuSPARSE [15], cuRAND [16], Thrust [17], and NVIDIA NPP [18].

B. *Compiler Directives*

In this method, compiler takes care of details of parallelism and data movement. The code is generic and can be deployed into multiple languages. The bad side of this method is that performance can vary across compiler versions. One of the popular examples is OpenACC. OpenACC is a user-driven directive-based performance-portable parallel programming model designed for scientists and engineers interested in porting their codes to a wide-variety of heterogeneous HPC hardware platforms and architectures with significantly less programming effort than required with a low-level model [19]. The OpenACC Application Programming Interface provides a set of compiler directives, library routines, and environment variables that can be used to write data parallel in Fortran, C/C++ programs that run on accelerator devices including GPUs and CPUs.

C. *Programming Languages*

This method is the best approach to developing GPU software. It enables programmers to have total control of parallelism and data movement. The computation does not need to fit into a limited set of library patterns or directives type. The approach needs the programmer to have sound knowledge about the development. Some of the GPU programming languages available are as follow: MATLAB, Mathematica, labView, CUDA Fortran, CUDA C/C++, pyCUDA, Copperhead, Numba, Alea.cuBase. This method is utilized in this paper with CUDA C

V. CUDA

Commune Unified Device Architecture (CUDA) is a comprehensive software and hardware architecture for GPGPU that was developed and released by NVIDIA in 2007. It is NVIDIA's move into GPGPU and High-Performance Computing (HPC), combining huge programmability, performance, and ease of use. A major design goal of CUDA is to support heterogeneous computations in a sense that serial parts of an application are executed on the CPU and parallel parts on the GPU [10]. CUDA platform works with programming languages like C, C++, and FORTRAN. This accessibility makes it easier for specialists in parallel programming to use GPU resources, in contrast to prior APIs like Direct3D and OpenGL, which required advanced skills in graphics programming [11]. CUDA also supports frameworks like OpenACC and OpenCL.

A. *CUDA Program Structure*

The CPU is also referred to as the HOST while the GPU is known as DEVICE. CUDA supports both serial and parallel implementations; the CPU and GPU take part in the compilation of the whole application. The parts that exhibit little or no data parallelism are implemented in the host code. The parts that have rich amount of data parallelism are implemented in the device code. A CUDA program is a unified source code encompassing both host and device code [20]. The NVIDIA® C compiler (NVCC) separates the two during the compilation process. The host code is straight ANSI C code; it is further compiled with the host's standard C compilers and runs as an ordinary CPU process. The device code is written using ANSI C extended with keywords for labelling data-parallel functions, called kernels, and their associated data structures. The device code is typically further compiled by the NVCC and executed on a GPU device. Fig. 2 shows detailed information how CUDA C program is being compiled.

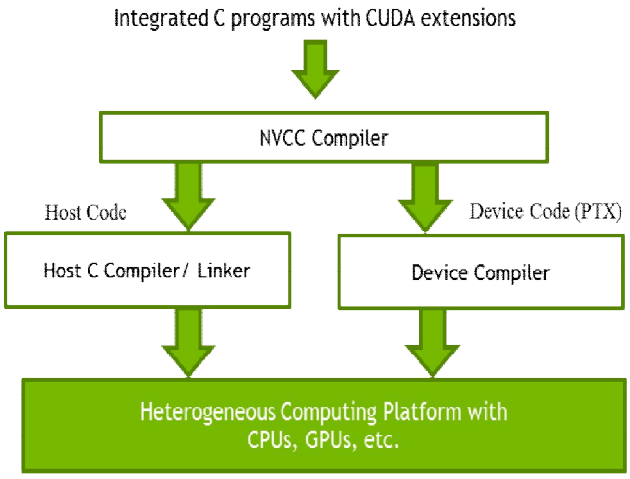


Fig. 2. Compiling a CUDA program [20]

The execution is illustrated in Fig. 3. The execution starts with host (CPU) execution. When a kernel function is invoked, or launched, the execution is moved to a device (GPU), where a large number of threads are generated to take advantage of abundant data parallelism. All the threads that are generated by a kernel during an invocation are collectively called a grid.

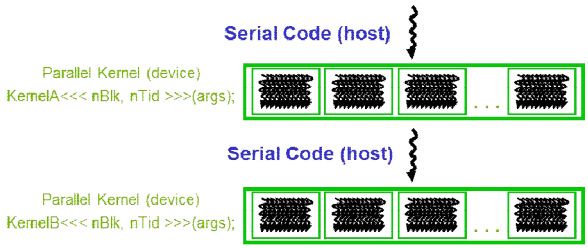


Fig. 3. Executing a CUDA program [20]

Fig3. shows the execution of two grids of threads. When all threads of a kernel complete their execution, the corresponding grid terminates, and the execution continues on the host until another kernel is invoked [20].

VI. Implementation

Linear Regression is calculated by finding the best-fitting straight line through the points, regression line. The formula for a regression line is:

$$Y' = bX + A$$

Where Y' is the predicted score, X is the given value, b is the slope of the line and A is the Y intercept.

The slope (b) can be calculated as follows:

$$b = r\, sY/sX$$

Where sX is the standard deviation of x, sY is the standard deviation of y and r is the coefficient of linear regression.

The intercept (A) can be calculated as:

$$A = MY - bMX$$

Where MY is the mean of Y and MX is the Mean of x.

All these parameters are computed in parallel to get the regression line which is then used to calculate the expected value of Y which is the expected sales. Y represents the quantity of sales while X represents the prices of sales.

A. Algorithm

To calculate the predicted value of Y (sales) given a particular price of goods, we need to implement the regression line programmatically. The first step is to calculate the mean of X and Y which will then be used to calculate the standard deviation of X and Y. The correlation coefficient (r) is calculated from the standard deviations of X and Y. The slope (b) is then calculated using the formula, b = r sy/sx. The intercept is also calculated and the expected value of Y is estimated afterwards. The steps below explain the flow of execution.

1) Load the value of X and Y into the Device Memory.
2) Use all threads to copy the values of X into shared memory.
3) Save two copies of X and Y separately in the shared memory.
4) Perform Reduction in the shared memory to get the sum of X.
5) Use one thread to get the mean of X, by dividing the sum with the number of elements.
6) Repeat the same process to calculate the mean of Y.
7) Use the same process to calculate the Mean deviation, Variance, and Standard deviation of X and Y.
8) Use one thread to calculate the slope, intercept and the expected value of Y.

B. Flowchart

Fig. 4 shows the flowchart for the CPU and GPU implementation of the algorithm. As shown below, GPU will calculate the sum of X and Y in parallel and use the value to get the mean of both variables, while CPU does this task serially. The standard deviations of both variables are also calculated in parallel for GPU and serially for CPU. We found out that GPU implementation performed effectively than CPU implementation.

VII. Results And Performance

The problem explored in this paper is to implement sales forecasting using linear regression on the GPU. We implemented both serial and parallel code and have run both algorithms using 10 data sets of sales data and we have measured the speedup (ratio of time of serial algorithm to that for parallel algorithm).

We have used Intel(R) Xeon(R) CPU E5-2683 v3 @ 2.00GHz and a GPU Tesla K20Xm. This card has 2,688 processor cores, 732 MHz processor clock and 249.6 GB/sec memory bandwidth.

Ten sales data with different data size were used to test the implementation as shown in Table I. Table I also shows the CPU, GPU elapsed executions times and the speed-up was achieved for each data. It was observed that as the size of sales data increases the speed-up also increases.

Fig. 5 shows the elapsed time of CPU and GPU solutions using the ten sales data of different sizes. As we see in Fig. 5 we can tell that the GPU approach is faster than the CPU approach.

International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-6, Issue-4, July-2018
DOI: 10.21276/ijircst.2018.6.4.1

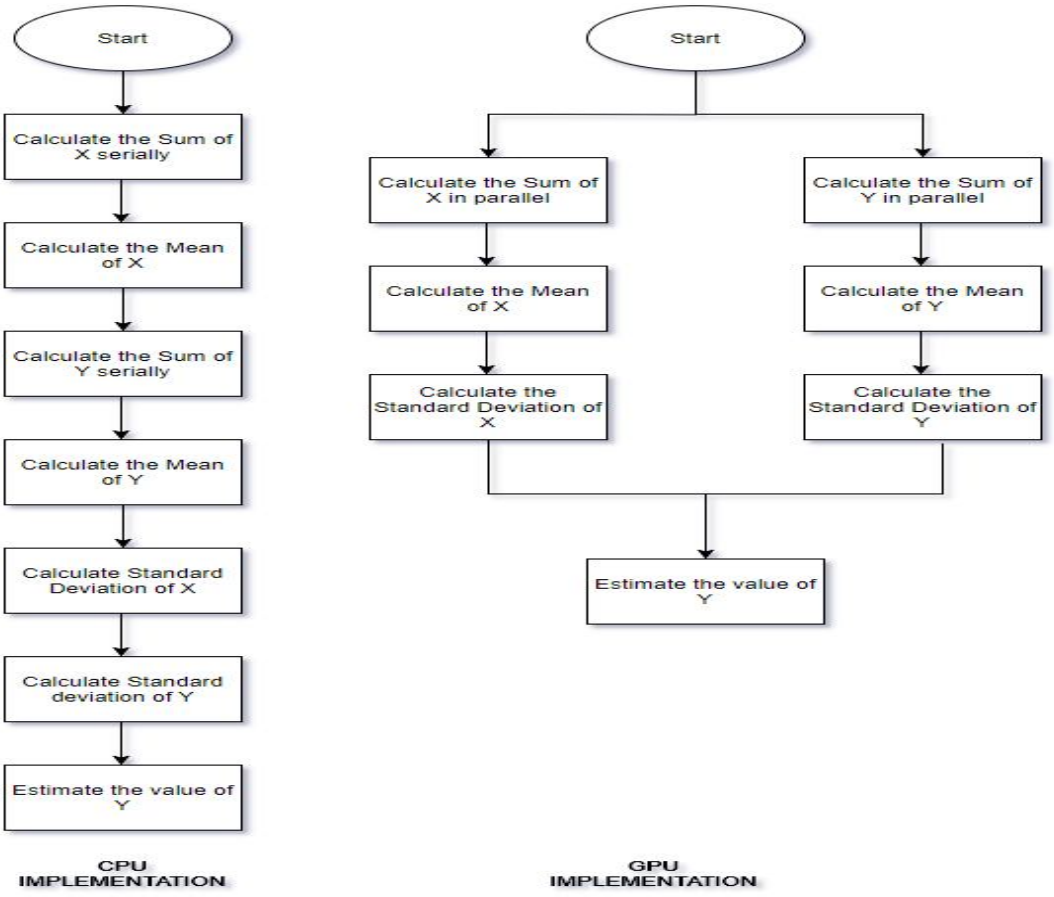


Fig. 4. Sales forecasting flowchart

TABLE I. COMPARISON OF CPU AND GPU EXECUTION TIME

S/No	No of Sales	CPU Time (ms)	GPU Time (ms)	Speed Up
1	200	0.529	0.706	0.749
2	400	1.058	0.645	1.640
3	600	1.587	0.646	2.456
4	800	2.116	0.670	3.158
5	1000	2.645	0.675	3.918
6	1200	3.174	0.680	4.667
7	1400	3.703	0.678	5.461
8	1600	4.232	0.690	6.133
9	1800	4.761	0.695	6.850
10	2000	5.29	0.700	7.557

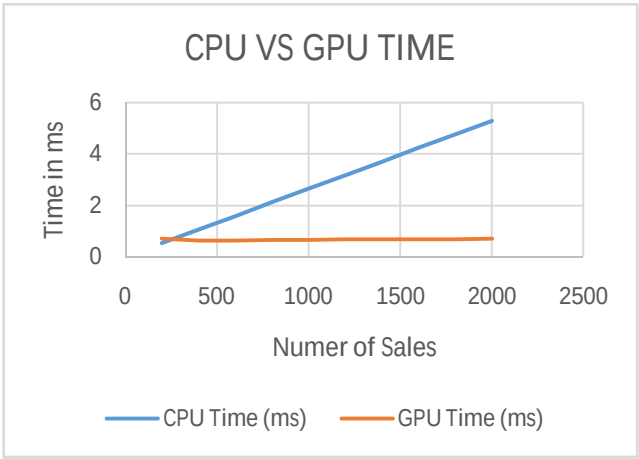


Fig. 5. CPU Vs GPU time

VIII. CONCLUSION AND FUTURE WORK

Sales Forecasting by linear regression on GPU using CUDA was implemented on NVIDIA Graphical Processing Unit (GPU) together with Central Processing Unit (CPU). It was observed that the multithreading architecture and SIMD approach of CUDA helps to speed up the performance of the process. There is significance difference

in the performance obtained on GPU. This research handles forecasting of one product at a time, a future work could be used to handle different products at the same time.

REFERENCES

[1] http://hbr.org/1971/07/how-to-choose-the-right-forecasting-technique

[2] Sales forecasting is not about complex algorithm.

[3] http://www.yourarticlelibrary.com/sales/sales-forecasting-top-9-methods-of-sales-forecasting/50998

[4] https://blog.getbase.com/5-essential-sales-forecasting-techniques

[5] https://en.wikipedia.org/wiki/Linear_regression

[6] Li Bing-jun, He Chun-hua, China, "The Combined Forecasting Method of GM(1,1) with Linear Regression and Its Application" 2007 IEEE International Conference.

[7] "Programming with CUDA", www.nvidia.com.

[8] Jyoti B. Kulkarni1, A. A. Sawant2, Vandana S. Inamdar3 Database Processing by Linear Regression on GPU using CUDA

[9] "Getting Started with CUDA", www.nvidia.com

[10] Shadi Alawneh and Dennis Peters, Proc. 14th IEEE International Conference on High Performance Computing and Communications (HPCC-2012), June 2012, Liverpool, UK.

[11] https://en.wikipedia.org/wiki/CUDA

[12]SamanehBeheshti-Kashi, Hamid Reza Karimi,Klaus-Dieter Thoben, Michael Lütjen and Michael Teucke, "A survey on retail sales forecasting and prediction in fashion markets, " Systems Science & Control Engineering, Volume 3, 2015.

[13] Henrik Aronsson and Rickard Jonsson, "Sales forecasting Management," A bachelor thesis in Management Accounting, 2008.

[14] Rashmi Sharma and Ashok K. Sinha, "Sales forecasting of an automobile industry," International Journal of Computer Applications

[15] http://docs.nvidia.com/cuda/cusparse/index.html

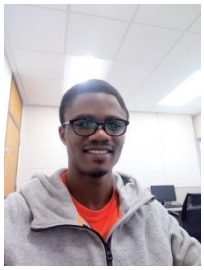
[16] http://docs.nvidia.com/cuda/curand/index.html

[17] http://docs.nvidia.com/cuda/thrust/index.html

[18]http://docs.nvidia.com/cuda/npp/index.html

[19]http://openacc.org

[20]David B. Kirk and Wen-mei W. Hwu, Programming Massively Parallel Processor.



Ayomide Yusuf is graduate student (Masters in Embedded Systems) at Department of Electrical and Computer Engineering in Oakland University. He had his Bachelor's Degree in Computer Science at University of Ilorin in Nigeria. He used Fuzzy Logic Methodology to develop an optimized algorithm for predicting rainfall in his undergraduate thesis. Ayomide's research interests are in Artificial Intelligence, Machine Learning, Embedded design with GPU, software optimization, software design analysis, and autonomous systems. Ayomide is an experienced software developer, having worked as a software developer for three years after his first degree.

Shadi Alawneh received the BEng degree in Computer Engineering from the Jordan University of Science and Technology, Irbid, Jordan in 2008, the MEng and PhD degrees in computer engineering from the Memorial University of Newfoundland, St. John's, NL, Canada, in 2010 and 2014, respectively. Then, he joined the Hardware Acceleration Lab at IBM Canada as a staff software developer from May 2014 through August 2014. After that, he joined C-CORE as a research engineer from 2014 until 2016 and became adjunct professor in the Department of Electrical and Computer Engineering at Memorial University of Newfoundland in 2016. Dr. Alawneh is currently an assistant professor in the department of Electrical and Computer Engineering at Oakland University. Dr. Alawneh has authored or co-authored scientific publications (including international peer-reviewed journals and conferences). His research interests include parallel and distributed computing, general purpose GPU computing, parallel processing architecture and applications, numerical simulation and modeling, software design and optimization. He is a member of the IEEE Computer Society.

