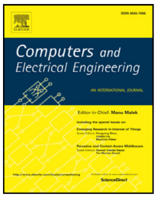
Computers and Electrical Engineering 000 (2017) 1–13

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng



An analysis framework for hardware and software implementations with applications from cryptography[☆]

Issam Damaj[a,*], Safaa Kasbah[b]

[a] *Electrical and Computer Engineering Department, American University of Kuwait, Salmiya, Kuwait*
[b] *Computer Science and Mathematics Department, Lebanese American University, Beirut, Lebanon*

ARTICLE INFO

Article history:
Received 1 June 2017
Revised 5 June 2017
Accepted 5 June 2017
Available online xxx

Keywords:
Analysis
Hardware
Software
Gate arrays
Algorithms
Cryptography

ABSTRACT

With the richness of present-day hardware architectures, tightening the synergy between hardware and software has attracted a great attention. The interest in unified approaches paved the way for newborn frameworks that target hardware and software co-design. This paper confirms that a unified statistical framework can successfully classify algorithms based on a combination of the heterogeneous characteristics of their hardware and software implementations. The proposed framework produces customizable indicators for any hybridization of processing systems and can be contextualized for any area of application. The framework is used to develop the Lightness Indicator System (*LIS*) as a case-study that targets a set of cryptographic algorithms that are known in the literature to be tiny and light. The *LIS* targets state-of-the-art multi-core processors and high-end Field Programmable Gate Arrays (*FPGAs*). The presented work includes a generic benchmark model that aids the clear presentation of the framework and extensive performance analysis and evaluation.

1. Introduction

With the advancements in high-performance computing, algorithms have a wide range of efficient implementation options. Current computers can be equipped with multi-core processors, Graphics Processing Units (*GPUs*), and high-end programmable devices, such as, *FPGAs*. The variety of processing options are supported by a wealth of co-design tools that facilitates hardware and software implementations [1,2]. Nevertheless, several questions remain on what algorithm is the best to suite an implementation option, and vice-versa. How would an algorithm perform within hybrid processing systems, and how to make an evaluation based on heterogeneous performance measurements?

The core of any performance measurement includes measures, metrics, and indicators. Indicators are defined as qualitative or quantitative factors, or variables that provide simple and reliable means to measure achievement. A qualitative performance indicator is a descriptive characteristic, an opinion, a property or a trait. However, a quantitative performance indicator is a specific numerical measurements resulted by counting, adding, averaging numbers or other computations [3]. Qualitative and quantitative measurements can be combined to define measurement frameworks and benchmarks [4]. There

[☆] Reviews processed and recommended for publication to the Editor-in-Chief by Area Editor Dr. G. Martinez Perez.
[*] Corresponding author.
E-mail addresses: idamaj@auk.edu.kw (I. Damaj), safaakasbah@gmail.com (S. Kasbah).

http://dx.doi.org/10.1016/j.compeleceng.2017.06.008
0045-7906/© 2017 Elsevier Ltd. All rights reserved.

is a large number of hardware and software benchmarks in the literature. Yet, limited research work is reported to address developing analysis frameworks for heterogeneous hardware and software implementations.

In this paper, we present a statistical analysis framework for performance profiling of related algorithms running under different hardware and software subsystems. The framework comprises criteria, indicators, and measurements obtained from heterogeneous sources. The measurements are statistically combined to produce indicators that capture the algorithmic, software, and hardware characteristics of the assessed algorithms. The developed framework enables the deep and thorough reasoning about each hardware and software subsystem, and combines heterogeneous characteristics to provide overall ratings, rankings, and classifications. The proposed framework is customizable for any hybridization of processing systems and can target any model of computation or area of application.

The paper includes the development of a generic benchmark model that serves as a specification pattern of analysis and evaluation frameworks. The model captures the activities, resources, implementation, mathematical formulation, and intended measurements of an analysis framework or a benchmark. The developed model can be used to describe any benchmark with simplicity and clarity. The model is adopted to present the proposed analysis framework.

To validate the proposed framework in application, a case-study is carried out with application from cryptography. The case-study enables the development of the*LIS* with its bouquet of statistical indicators. The *LIS* formulates the proposed framework within the context of lightweight cryptographic algorithms. The proposed performance analysis classifies the investigated algorithms into a combination of their mathematical, software, and hardware characteristics. The two main targeted high performance computing devices are multi-core processors for software implementations and *FPGAs* for hardware implementations.

The rest of the paper is organized so that Section 2 surveys the literature. In Section 3, the motivation, research questions, and the paper contribution are presented. In Section 4, the generic benchmark model and the analysis framework are presented. Section 5 introduces the *LIS* according to the generic model. A thorough performance analysis and evaluation is presented in Section 6. Section 7 concludes the paper and sets the ground for future work.

2. Related work

2.1. Benchmarks

Benchmarks are widely addressed in the literature. Famous benchmarks include Whetstone, LINPAC, Dhrystone, Standard Performance Evaluation Corporation (SPEC), etc [5–7]. Several developments of embedded systems Benchmarks are lead by the Embedded Microprocessor Benchmark Consortium (*EEMBC*). *EEMBC* helps system designers in selecting the optimal processors, smartphones, tablets, and networking appliances. *EEMBC* mainly targets embedded system's hardware and software. *EEMBC* organizes its benchmark suites targeting automotive, digital media, multi-core processors, networking, automation, signal processing, hand-held devices, and browsers. The benchmarks developed by *EEMBC* include *AutoBench, BrowsingBench, AndBench*, and *MiBench* [8]. Cryptography benchmarks are designed to measure the performance of different cryptographic algorithms running under different systems, such as, *GPUs* or other processors. Rukhin et al. in [9] presents a statistical test suite for random and pseudorandom number generators for cryptographic applications. Yue et al. in [10] presents a cryptographic benchmark suite for network processors (NPCryptBench).

2.2. Hardware/software co-design evaluation frameworks

Performance analysis and evaluation within hardware/software (HW/SW) co-design investigations are usually based on a variety of metrics. Besides standard metrics, such as execution time, maximum frequency, throughput, hardware resource utilization, power consumption, etc., several metrics are identified within the context of application. Jain–Mendon and Sass in [11] proposed a HW/SW co-design approach for implementing sparse matrix vector multiplication on *FPGAs*. Within the context of application, the authors evaluated their approach by analyzing the hardware and software implementations in terms of the speed of processing floating point operations, bandwidth efficiency, data block size, communication time, etc. Lumbiarres–Lopez et al. [12] implemented, within a co-design environment, a countermeasure against side-channel analysis attacks. The used application-specific metrics comprise the difference in change of input current over time and correlations between data and power consumption. All the aforementioned investigations employed the standard co-design metrics. In [13], the performance of block tridiagonal solvers was evaluated under heterogeneous multi-core processors and *GPUs*. The evaluation was mainly based on analyzing memory performance and measuring the total execution times of different scenarios.

The standard metrics of co-design applies to partitioned hardware and software implementations. The focus in partitioned implementation is the analysis and evaluation of the developed subsystems with an aim to find the best possible partitioning strategy. Wu et al. [14] studied the performance and algorithmic aspects of a proposed heuristic partitioning algorithm. The produced implementations were analyzed with-respect-to execution time, resource utilization, and the attained solution quality as related to the smallest possible error. In [15], Jemai and Ouni proposed a partitioning strategy based on control data graphs. The partitioning algorithm was deployed within three different case-studies. The metrics analyzed across the three studies comprised the number of partitions, software execution time, hardware resource utilization,

Please cite this article as: I. Damaj, S. Kasbah, An analysis framework for hardware and software implementations with applications from cryptography, Computers and Electrical Engineering (2017), http://dx.doi.org/10.1016/j.compeleceng.2017.06.008