2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines

Accelerating Big Data Analytics Using FPGAs

Katayoun Neshatpour[1], Maria Malik[1], Mohammad Ali Ghodrat[2], and Houman Homayoun[1]

[1]Department of Electrical and Computer Engineering, George Mason University
[2]Department of Computer Science, University of California, Los Angeles

I. Introduction

Emerging big data analytics applications require a significant amount of server computational power. As chips are hitting power limits, computing systems are moving away from general-purpose designs and toward greater specialization. Hardware acceleration through specialization has received renewed interest in recent years, mainly due to the dark silicon challenge. To address the computing requirements of big data, and based on the benchmarking and characterization results, we envision a data-driven heterogeneous architecture for next generation big data server platforms that leverage the power of field-programmable gate array (FPGA) to build custom accelerators in a Hadoop MapReduce framework. Unlike a full and dedicated implementation of Hadoop MapReduce algorithm on FPGA, we propose the hardware/software (HW/SW) co-design of the algorithm, which trades some speedup at a benefit of less hardware. Considering communication overhead with FPGA and other overheads involved in Hadoop MapReduce environment such as compression and decompression, shuffling and sorting, our experimental results show significant potential for accelerating Hadoop MapReduce machine learning kernels using HW/SW co-design methodology.

II. System Architecture and Methodology

Most recent works on big data acceleration focus on the implementation of an entire particular machine learning application or offloading phases of its MapReduce to the FPGA hardware, which results in excessive hardware and extensive design effort. Alternatively, we propose a HW/SW co-design of the machine learning algorithm implemented in Hadoop MapReduce, which trades some speedup at a benefit of less hardware. Fig. 1 shows the heterogeneous architecture, which consists of a high-performance CPU as the master node connected to several Zynq devices as slave nodes. Various communication overheads between master and slaves have been marked with "OV" in Fig.1.

The master node runs the HDFS and is responsible for the job scheduling between all the slave nodes. Intel Atom C2758 and Xeon E5 were used as the master node. Xeon cores are mostly designed for high-performance servers, and Atom cores advocate the use of a low-power core to address the dark silicon challenge facing servers. The slave nodes are Zynq SoCs, which consist of ARM processors coupled with programmable logic.

We study four widely used data mining and machine learning algorithms, K-means, KNN, SVM and Naive Bayes, to find the CPU-intensive and time-consuming kernels (hotspot functions) to offload to FPGA and calculate the kernel speedup through HW/SW co-design. It is important to notice that, in an end-to-end Hadoop platform, not all the execution time is dedicated to the kernel. Hadoop file system management, compression and decompression,

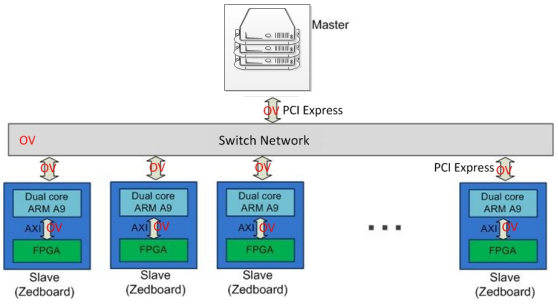

Figure 1. System architecture.

shuffling and sorting, and standard Java library access are performed at GHz clock frequency on CPU server. Based on Intel Vtune profiling results, we find the ratio of execution time that corresponds to the kernel. Consequently, Amdahl's law is utilized to understand how the speedup of the kernel contributes to its overall execution in an end-to-end Hadoop MapReduce environment.

III. Implementation Results

Fig. 2 shows the best speedup results for the applications in a MapReduce framework with one mapper slot. Fig. 2 shows how the speedup of each application through HW/SW acceleration is translated into a lower speedup on the an end-to-end Hadoop system, considering all communication overhead between master and slaves and all the tasks involved in MapReduce environment. For instance, while the acceleration of the K-means yields a speedup of $312\times$ with zero overhead, the speedup is reduced to $146\times$ with the HW/SW data transfer overhead, and $2.72\times$ and $2.78\times$ on Hadoop platform with Atom and Xeon, respectively.

It should be noted that various design aspects including the size of input data, size of splits to the mapper functions, number of mapper and reducer slots, type of master core, memory configuration, etc., influence the performance, power and energy efficiency. These aspects have to be evaluated in order to find the optimal design for each application.

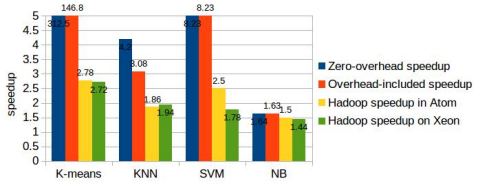

Figure 2. Acceleration on a full-blown Hadoop platform.

978-1-4799-9969-9/15 $31.00 © 2015 IEEE
DOI 10.1109/FCCM.2015.59