Acceleration of Data Streaming Classification Using Reconfigurable Technology

Pavlos Giakoumakis(✉), Grigorios Chrysos,
Apostolos Dollas, and Ioannis Papaefstathiou

Microprocessor and Hardware Laboratory Department of Electronic and Computer Engineering, Technical University of Crete, Chania, Greece
{pgiakoumakis,gregory.chrysos}@gmail.com,
{dollas,ygp}@ece.tuc.gr

Abstract. The challenging task for streaming data is to store, analyze and visualize massive volumes of data using systems with memory and run-time constraints. This paper presents a novel FPGA-based system to accelerate the classifier building process over data streams. The proposed system maps efficiently the well-known Very Fast Decision Tree (VFDT) streaming algorithm on a single custom processor in a high-end Convey HC-2ex FPGA platform. The experimental results show that the proposed system outperforms by two orders of magnitude the software-based solutions for streaming data processing, whereas the available resources of the platform allow for substantial further scaling.

Keywords: Data Stream Mining · Machine Learning · Decision Trees · VFDT · Hardware accelerator · FPGA

1 Introduction

Machine Learning (ML) is a research area of computer science that focuses on the extraction of useful information from raw data. The ML algorithms process data either in streaming mode or batch mode. The batch mode processing, which is driven largely by platforms like Apache Hadoop, provides mechanisms for running analytics across massive volumes. On the other hand, the streaming processing platforms need to cover two main assumptions: the reduced access to the data, which can be "seen" only once, and the time and memory restrictions during the processing stage. There are several widely used stream processing engines, like Storm and S4.

This work presents a novel FPGA-based architecture for streaming data processing. The implemented architecture maps a well-known classification streaming algorithm, i.e. the Very Fast Decision Tree (VFDT) [1] algorithm, on a reconfigurable platform with very promising performance results. The contributions of this work are:

- This is the first, to the best of our knowledge, hardware-based architecture to build the VFDT data streaming classifier on a high-end FPGA platform.
- Our proposed architecture takes advantage of the fine-grained and the coarse-grained parallelism that reconfigurable hardware can offer vs. the software coarse grained parallelism.

K. Sano et al. (Eds.): ARC 2015, LNCS 9040, pp. 357–364, 2015.
DOI: 10.1007/978-3-319-16214-0_31

- The single-FPGA-based single processor execution unit, which was implemented, achieves at least two orders of magnitude higher processing data rates vs. the execution of the VFDT algorithm on a single-thread platform.

The rest of this paper is organized as follows: Section 2 presents prior software and hardware-based works for data stream processing. Section 3 describes the streaming classification problem and analyzes the VFDT algorithm. Section 4 presents the architecture of the implemented system. Section 5 evaluates the performance of the proposed system. Finally, Section 6 draws the conclusions of this work.

2 Parallel Data Classification

Data classification is a well-studied problem and many software and hardware solutions have been proposed.

2.1 Software-Based Data Classification Platforms

There are many software-based platforms that focus on acceleration of the data classification workload. The Planet project [2] and the OpenPlanet project [3] used a cluster of processors to achieve linear reduction in the training execution time of the classification workload. Moreover, He *et al.* in [4] presented parallel implementations of various classification algorithms based on Hadoop's MapReduce process to achieve linear speedup vs. the size of the input dataset. Lastly, Tyree *et al.* [5] presented a parallel implementation of the gradient boosted regression tree algorithm using the MapReduce framework to achieve one order of magnitude performance speedup vs. single thread solution.

On the other hand, the most widely used software-based framework for data stream mining is the Massive On-line Analysis (MOA) framework [6]. Another important stream processing tool is the Scalable Advanced Massive Online Analysis (SAMOA) [7] framework by the Yahoo Labs.

2.2 Hardware-Based Data Classification Systems

This section presents some previous reconfigurable computing-based works that focused on data classification. Luo *et al.* [8] presented an FPGA-based architecture implementing a fast Decision Tree Searching for IP Traffic Classification. Papadonikolakis *et al.* [9] proposed an FPGA-based architecture for Support Vector Machine (SVM) classification and their system achieved 2–3 orders of magnitude as compared to the corresponding CPU implementation. Narayanan *et al.* [10] proposed an FPGA-based architecture that implements the most compute-intensive kernel of most DTC algorithms. Their implementation offers a 5.5x performance speedup vs. the software implementation. Chrysos *et al.* [11] presented the HC-CART system, which is an FPGA-based system that implements the Classification and Regression Tree algorithm on a multi-FPGA platform, achieving performance speedup up to two orders of magnitude vs. single threaded solutions, together with exceptional power efficiency.

Lastly, the issues involved in Big Data processing on present-day FPGA supercomputing platforms are presented by Dollas in [12].

This paper is the first work to map the widely used VFDT algorithm on hardware and the first hardware-based work on real-time machine learning streaming problems.

3 Classification

Classification is the process of assigning unseen instances to categories, based upon an existing model. The classification process is divided into two phases: model building and model testing.

3.1 VFDT Algorithm

The decision tree classification uses a decision tree as a predictive model for addressing the classification problem. The VFDT algorithm's trained model is a decision tree structure with a dynamic incremental nature. It begins with a single node, i.e. the root, and each time a new data instance arrives it calculates some functions, like the Information Gain or the Gini index, and it updates node statistics. Next, it finds the splitting rules that lead to the two best split decisions. Lastly, it uses the Hoeffding bound to decide if it is beneficial to apply the best split suggestion or not. The process continues recursively each time a new data instance arrives. The process is presented in Fig. 1.

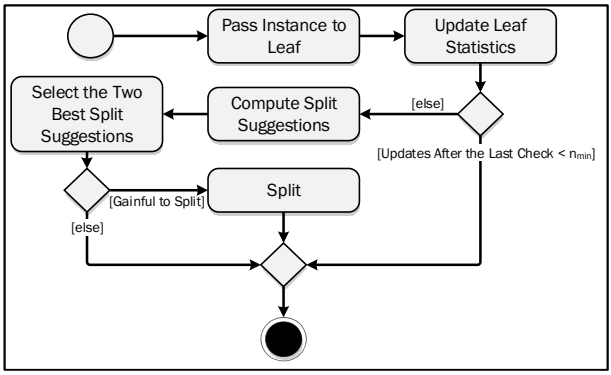

Fig. 1. Building of streaming classification model

3.2 VFDT Algorithm Analysis

This section presents some important issues as far as the nature of the VFDT algorithm. The algorithm analysis showed that:

- The training process is a much more computationally intensive process than the classification one.

- The most time consuming part of the algorithm is the computation of the split suggestions at the leaves. Thus, we focused on accelerating the leaf operations by mapping these operations on reconfigurable technology.
- The available parallelism of the problem lies into three directions.
  - Parallel traversing of the tree. Each new instance traverses the model-tree independently from the previous or the next data instance.
  - Parallel leaf processing. The operations at the model's leaves can take place in parallel.
  - Parallel attribute processing. Operations at the model's leaves such as updating the leaf statistics or the computation of split suggestions could be split into smaller parts that can take place in parallel for different attributes.

4 Architecture

This section presents the proposed architecture for mapping such compute intensive and streaming workloads on a reconfigurable platform, such as the Convey HC-2ex.

4.1 Attribute Processing Module

The Attribute Processing Module is the core unit of the proposed architecture, which is presented in Fig. 2. The proposed unit is used to update the attribute statistics and to compute the best attribute splitting rules. The Attribute Processing Module consists of two basic subunits: the Update Cache unit and the Compute Best Split Suggestion unit.

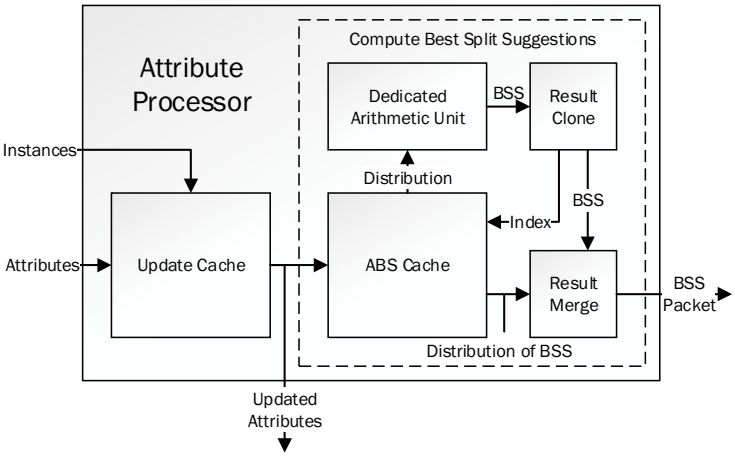

Fig. 2. Attribute Processing Module

The Update Cache module updates the statistics of a single tree node depending on the data instances that arrive. As shown in Fig. 2, it takes the instances that arrive to a leaf node and the statistics that are kept in these nodes. Then it updates the data and

sends them back to the memory for storage and to the next module to compute the best split rule.

The Compute Best Split Suggestion unit takes as input the updated statistics and calculates the best splitting rule for the processed leaf node. The ABS Cache module and the Dedicated Arithmetic Unit implement the main workload for calculating the best attribute split rule.

4.2 Extended Attribute Processing Array

The Extended Attribute Processing Array is the basic unit of our proposed architecture. Our proposed solution communicates via a fast I/O data link with the corresponding software host part, as shown in Fig. 3. The proposed module consists of two basic subunits: the Attribute Processing Array (APA) and the Memory Controller Management Unit (MCMU).

The Attribute Processing Array, i.e. APA, consists of a series of parallel Attribute Processing Modules. The DMA modules residing in MCMU feed APA with attributes and instances while the dispensers deliver the data to the Attribute Processors. Each Attribute Processor outputs the updated statistics of a leaf node and the best split suggestion for each one of the attributes, which are sent back to the memory via the Updated Attribute and the Best Split Suggestion (BSS) Multiplexers. Lastly, the Selection Unit compares all the output best splitting rules and keeps the best splitting rule from all the input attributes.

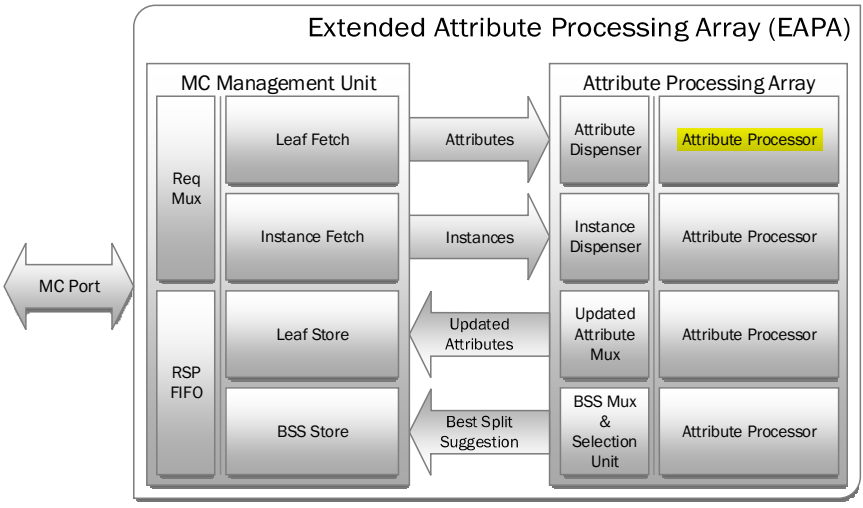

Fig. 3. Extended Attribute Processing Array (EAPA)

The Memory Controller Management Unit is responsible for data I/O from the shared memory. The memory requests are multiplexed by the Request Multiplexer. All four memory-based engines, i.e. Leaf fetch, Instance Fetch, Leaf store and BSS store modules, are designed to use 8-byte bursts. Furthermore, the MCMU is designed

to take full advantage of the 64-bit memory controller port that runs at the same frequency with it.

The EAPA module is designed to be a firm basis for a larger architecture, where parallel EAPA modules can be employed to take advantage of more of the parallelism between attributes.

5 System Evaluation

This section presents the evaluation of the implemented system as far as the resources that were used and the performance results that were achieved.

5.1 Resources

An instance of our proposed architecture was implemented on a Convey HC-2ex server. Our system uses only about 5% of the available resources of a Virtex 6 LX760 FPGA device (out of four in the platform). Thus, our system can be easily parallelized in the future with many EAPAs on a single FPGA device, and then with multiple FPGAs in the system – as long as the memory controller can provide data. Our implemented system runs at 150 MHz, i.e. the clock rate for the memory data I/O ports.

5.2 Performance

This section presents the performance achieved by our proposed system vs. the single-thread Java-based implementation of the Hoeffding Tree Classification algorithm from MOA tool version 2014.4[6]. The software experiments took place on the host processor of the Convey HC-2ex platform, which includes a 4-core Intel Xeon CPU @ 2.13 GHz with 24GB RAM. We used various input datasets for our system evaluation. We built random datasets varying the parameters of the input dataset, i.e. the number of attributes per input instance, the number of classes per input instance and the number of values per attribute. The performance results from our three different groups of experiments are presented in the following Tables.

Table 1. Experiment with variable number of attributes (1 instance per leaf, 1024 leaves, 8 classes, 256 values per attribute)

No. of attributes	MOA Hoeffding Tree Classification Execution Time (sec)	FPGA-based Hoeffding Tree Classification Execution Time (sec)	Speedup
32	73.84	0.51	145x
64	147.89	0.96	154x
128	295.54	1.85	160x
256	590.95	3.63	163x
512	1182.55	7.19	164x
1024	2363.11	14.31	165x

Table 2. Experiment with variable number of classes (1 instance per leaf, 1024 leaves, 1024 attributes, 256 values per attribute)

No. of classes	MOA Hoeffding Tree Classification Execution Time (sec)	FPGA-based Hoeffding Tree Classification Execution Time (sec)	Speedup
2	638.11	3.61	177x
4	1211.46	7.18	169x
6	1785.15	10.75	166x
8	2361.91	14.31	165x

Table 3. Experiment with variable number of attributes (1 instance per leaf, 1024 leaves, 8 classes, 1024 attributes)

No. of values per attribute	MOA Hoeffding Tree Classification Execution Time (sec)	FPGA-based Hoeffding Tree Classification Execution Time (sec)	Speedup
32	79.02	1.83	43x
64	219.62	3.61	61x
128	673.73	7.17	94x
256	2361.44	14.31	165x

The above results show that our proposed architecture offers up to two order of magnitude performance speedup vs. a software solution that is implemented in a widely used tool, the MOA tool. Despite the streaming nature of the problem we showed different ways to parallelize the problem using hardware techniques like pipelining and fine-grained parallelization. In addition to this, we showed that the nature of the input datasets does not influence the performance of our proposed design. Lastly, the most important issue is that our system offers multiple distinct parallelization levels on data processing, which cannot be leveraged by processors in software.

6 Conclusions

This work presented an FPGA–based architecture that implements a widely used classification data streaming algorithm, the VFDT algorithm. Our proposed architecture combines fine-grained and coarse-grained parallelization level, which can be efficiently mapped on reconfigurable hardware. Also, we showed that the ensemble of modules that cooperate in a pipelined way to process a continuous stream of data seems to be a good architecture for such data streaming problems. Our proposed architecture was mapped and fully implemented on a Convey HC-2ex platform, which seems to be a good solution for data streaming problems, due to its high I/O data transmission rate and its memory organization. The proposed FPGA-based system offers substantial performance advantages vs. widely used software solutions. Future work will focus on massive parallelism on all four Convey HC-2ex FPGAs and full exploitation of the memory-FPGA bandwidth.

Acknowledgment. This work was funded by the General Secretariat of Research and Technology of Greece (GSRT) under the project "AFORMI- Allowing for Reconfigurable Hardware to Efficiently Implement Algorithms of Multidisciplinary Importance", funded in the call "ARISTEIA" of the framework "Education and Lifelong Learning" (code 2427).

References

1. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 71–80. ACM, August 2000
2. Panda, B., Herbach, J.S., Basu, S., Bayardo, R.J.: Planet: massively parallel learning of tree ensembles with mapreduce. Proceedings of the VLDB Endowment **2**(2), 1426–1437 (2009)
3. Yin, W., Simmhan, Y., Prasanna, V. K.: Scalable regression tree learning on Hadoop using OpenPlanet. In: Proceedings of Third International Workshop on MapReduce and its Applications Date, pp. 57–64. ACM, June 2012
4. He, Q., Tan, Q., Ma, X., Shi, Z.: The High-Activity Parallel Implementation of Data Preprocessing Based on MapReduce. In: Yu, J., Greco, S., Lingras, P., Wang, G., Skowron, A. (eds.) RSKT 2010. LNCS, vol. 6401, pp. 646–654. Springer, Heidelberg (2010)
5. Tyree, S., Weinberger, K. Q., Agrawal, K., Paykin, J.: Parallel boosted regression trees for web search ranking. In: Proceedings of the 20th International Conference on World Wide Web, pp. 387–396. ACM, March 2011
6. Bifet, A., Holmes, G., Pfahringer, B., Read, J., Kranen, P., Kremer, H., Jansen, T., Seidl, T.: MOA: A Real-Time Analytics Open Source Framework. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011, Part III. LNCS, vol. 6913, pp. 617–620. Springer, Heidelberg (2011)
7. De Francisci Morales, G.: SAMOA: A platform for mining big data streams. In: Proceedings of the 22nd International Conference on World Wide Web Companion, pp. 777–778. International World Wide Web Conferences Steering Committee, May 2013
8. Luo, Y., Xiang, K., Li, S.: Acceleration of decision tree searching for IP traffic classification. In: Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, pp. 40–49. ACM, November 2008
9. Papadonikolakis, M., Bouganis, C. S., Constantinides, G.: Performance comparison of GPU and FPGA architectures for the SVM training problem. In: International Conference on Field-Programmable Technology, FPT 2009, pp. 388–391. IEEE, December 2009
10. Narayanan, R., Honbo, D., Memik, G., Choudhary, A., Zambreno, J.: An FPGA implementation of decision tree classification. In: Design, Automation & Test in Europe Conference & Exhibition, DATE 2007, pp. 1–6. IEEE, April 2007
11. Chrysos, G., Dagritzikos, P., Papaefstathiou, I., Dollas, A.: HC-CART: A parallel system implementation of data mining classification and regression tree (CART) algorithm on a multi-FPGA system. ACM Transactions on Architecture and Code Optimization (TACO) **9**(4), 47 (2013)
12. Dollas, A.: Big Data Processing with FPGA Supercomputers: Opportunities and Challenges. In: Proceedings of the IEEE Computer Society Annual Symposium on VLSI. ISVLSI, pp. 474–479. IEEE, July 2014