Multi-GPU approach for big data mining - global induction of decision trees

Krzysztof Jurczuk
Bialystok University of Technology, Poland
k.jurczuk@pb.edu.pl

Marcin Czajkowski
Bialystok University of Technology, Poland
m.czajkowski@pb.edu.pl

Marek Kretowski
Bialystok University of Technology, Poland
m.kretowski@pb.edu.pl

ABSTRACT

This paper identifies scalability bounds of the evolutionary induced decision trees (DT)s. In order to conquer the barriers concerning the large-scale data we propose a novel multi-GPU approach. It incorporates the knowledge of the global DT induction and EA parallelization. The search for a tree structure and tests is performed sequentially by a CPU, while the fitness calculations are delegated to GPUs, thus the core evolution is unchanged. The results show that the evolutionary induction is accelerated several thousand times by using up to 4 GPUs on datasets with up to 1 billion objects.

CCS CONCEPTS

• **Computing methodologies** → *Parallel algorithms*; *Classification and regression trees*;

KEYWORDS

evolutionary data mining, big data, decision trees, scalability bounds, parallel computing, graphics processing unit (GPU), CUDA

ACM Reference Format:
Krzysztof Jurczuk, Marcin Czajkowski, and Marek Kretowski. 2019. Multi-GPU approach for big data mining - global induction of decision trees. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3322045

1 INTRODUCTION

Decision trees (DTs) [4] are one of the most useful supervised learning methods for classification. A typical induction algorithm is a top-down approach that is based on the classical divide and conquer schema. The main consequences of locally optimal choices made in each tree node during the induction are overgrown DT classifiers. Emerging alternatives to the greedy top-down approaches include primarily evolutionary algorithms. Their global approach limits the negative effects of locally optimal decisions as a tree structure, tests in internal nodes and predictions in leaves are searched simultaneously [5]. However, evolutionary tree induction is much more computationally demanding [1].

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3322045

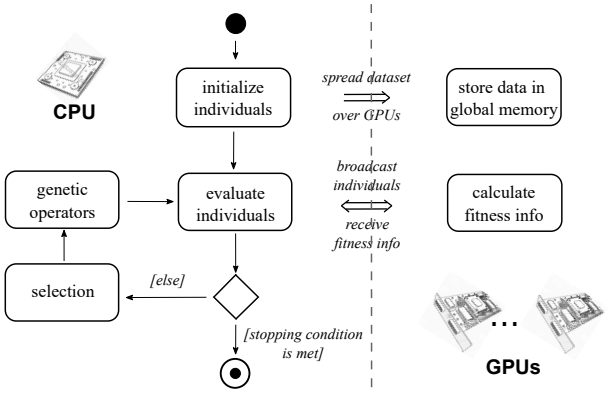

Figure 1: General flowchart of a GPU-accelerated approach.

To address this issue, we propose a novel multi-GPU approach with a data-parallel decomposition strategy in which the main loop (selection, genetic operators, etc.) is performed sequentially on a CPU, while the fitness calculations are parallelized on GPUs. It is applied to a system called Global Decision Tree (GDT) [3], which can be used for the evolutionary induction of various classification, regression and model trees. Experimental results indicate stunning improvement in the speed of the algorithm and the ability to perform big data mining.

2 MULTI-GPU APPROACH

In this paper, the multi-GPU approach is applied to a univariate binary classification tree from the GDT framework which follows a typical EA schema with an unstructured, fixed size population and a generational selection (see Figure 1). Individuals in the population are represented and processed in their actual form as classification trees with univariate tests in the internal nodes [5]. A fitness function is based on a simple weight formula that minimizes the prediction error and the tree complexity at the same time.

In the proposed approach, a CPU controls the evolutionary induction and performs relatively fast operations: population' initialization, selection, genetic operators (see Figure 1). The most time-consuming tasks (fitness calculation and dipoles searching) are isolated and delegated to GPUs. Such an architecture of the implementation ensures that the parallelization does not affect the behavior of the original EA. The first interaction between the CPU and the GPUs takes place during the initialization when the whole dataset is spread over the GPUs (data decomposition strategy) (see Figure 1). This CPU-GPUs transfer is done only once, each GPU

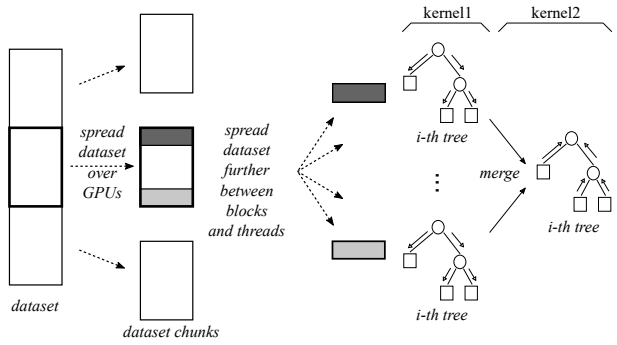

Figure 2: Work spreading over GPUs, blocks and threads.

receives an equal part of the dataset and the data is kept at the GPUs till the evolutionary inductions stops.

In the evolutionary loop, each time, when the genetic operator is successfully applied, the GPUs are called to help in the fitness evaluation and dipoles searching. For these purposes, all objects in the training dataset need to be passed through the tree starting from the root node to appropriate leaves. All GPUs concern the same individual in parallel but process different chunks of the dataset. Inside each GPU the dataset is divided further to spread dataset objects over GPU cores and, thus, providing a next level of parallelism (see Figure 2).

Calculations are divided into two kernel functions (*kernel1* and *kernel2*. The first one propagates objects from the tree root to the leaves. Each block works independently as it makes a copy of the evaluated individual that is later loaded into the shared memory. Threads process the same individual in parallel using different chunks of the data. At the end of this kernel, each GPU block stores in each tree leaf the number of objects of each class from the assigned part of the data that reaches the leaves. In addition, two objects of each class are randomly selected in each tree leaf. These objects will constitute mixed dipoles (pairs of objects of different classes). Such pairs may be easily used in some variants of genetic operators by the CPU, e.g. to find effective tests.

The *kernel2* function merges information from multiple copies of the individual allocated in each GPU block: objects' location in the leaves, class distributions and classification errors, starting from the leaves towards the root node. Obtained tree statistics and two objects (of each class in all tree nodes), randomly selected from ones provided by the first kernel, are sent back to the CPU.

The CPU when receiving a part of the results merges it with the overall result that are finally used to update the affected individual.

Table 1: Mean speedup (in case of 4 GPUs also mean execution time is included).

Dataset size	OpenMP	1 GPU	2 GPUs	4 GPUs	
1 000 000	5.8	519	672	936	(<0.5 min)
5 000 000	5.1	602	1165	2151	(≈ 1 min)
10 000 000	5.57	626	1270	2473	(≈ 2 min)
50 000 000	4.68	679	1265	2745	(≈ 10 min)

By default, for each GPU a separate CPU thread is created. When the CPU part is run on a multi-core/multi-processor hardware, parallel communication and data transfer is provided. Synchronization between parallel CPU threads is only needed when they merge partial results. To provide dataset size scalability the CPU does not have access to the objects except a few that constitute dipoles.

3 EXPERIMENTS

Experimental validation was performed on artificially generated dataset called *chess* that has objects arranged on a 3×3 chessboard with 2 real-value attributes [5]. We used a workstation equipped with 2 processors Intel Xeon E5-2620 v4 (20 MB Cache), 256 GB RAM, and 4 NVIDIA Tesla P100 GPU cards. Each CPU contained 8 physical cores running at 2.10 GHz. Each GPU card has 3584 CUDA cores and 12 GB of memory. A default, recommended set of GDT parameters were used [3, 5].

Table 1 shows the obtained mean speedup. The multi-GPU approach accelerates the tree induction at least hundreds times. For 50 millions of objects, the speedup almost reaches 3 000×. With 4 GPUs we manage to successfully induce a global decision tree for a 12 GB dataset with 1 billion of instances in approximately 4 hours which would take, we estimate, over a year for the sequential GDT system.

We see the huge gap between OpenMP (using 16 CPU cores) [2] and GPU parallelized versions. The scale of improvement of using multiple-GPUs depends on the size of the datasets, with larger datasets a super linear speedup is observed. With smaller datasets multiple GPU cores may not be fully saturated and other algorithm parts like GPU allocation/deallocations, the CPU calculations start to contribute a significant amount of time.

4 CONCLUSIONS

This papers shows the preliminary works on the scalability bounds of the evolutionary induced decision trees in context of big data. Our novel multi-GPU parallelization provides a significant speedup (even up to 3 000×). It scales linearly over GPUs providing a solution for billions of objects in hours. In future works, we plan to test it more thoroughly, apply some optimizations mechanisms as well as to extend it on the rest of GDT framework variants of decision trees including model trees.

Acknowledgments. This work was supported by the the Grants S/WI/2/18 (1st, 3rd author) and MB/WI/1/2017 (2nd author) from Bialystok University of Technology founded by Ministry of Science and Higher Education.

REFERENCES

[1] Rodrigo C Barros, Márcio P Basgalupp, Andre CPLF De Carvalho, and Alex A Freitas. 2012. A survey of evolutionary algorithms for decision-tree induction. *IEEE Transactions on SMC, Part C* 42, 3 (2012), 291–312.
[2] Marcin Czajkowski, Krzysztof Jurczuk, and Marek Kretowski. 2015. A Parallel Approach for Evolutionary Induced Decision Trees. MPI+OpenMP Implementation. In *Artificial Intelligence and Soft Computing (LNCS)*, Vol. 9119. Springer, 340–349.
[3] Krzysztof Jurczuk, Marcin Czajkowski, and Marek Kretowski. 2017. Evolutionary induction of a decision tree for large-scale data: a GPU-based approach. *Soft Computing* 21, 24 (2017), 7363–7379.
[4] S. B. Kotsiantis. 2013. Decision trees: A recent overview. *Artificial Intelligence Review* 39, 4 (2013), 261–283.
[5] Marek Kretowski. 2019. *Evolutionary Decision Trees in Large-Scale Data Mining*. Springer.