PERFORMANCE ENHANCEMENT OF WEKA: THE DATA MINING TOOL

Divya Jose*[1], Thomas George[2]

[1]M. Tech Student, Dept. of Computer Science, Jyothi Engineering College, Jyothi Hills, Vettikkattiri, Cheruthuruthy, Thrissur, Kerala, India.

[2]Asst. Prof, Dept. of Computer Science, Jyothi Engineering College, Jyothi Hills, Vettikkattiri, Cheruthuruthy, Thrissur, Kerala, India.

ABSTRACT

In the new area of data intensive science, data mining techniques receiving more attention. Nowadays, data analysts can rely on a broad number of tools, ranging in functionality, scope and target computer architectures. Data mining is an analytic process designed to explore data in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data Data mining tools is computationally demanding, so there is an increasing interest on parallel computing strategies to enhance their performance. The importance of Graphics Processing Units (GPUs) increased the comput- power of current desktop computers, but in desktop based data mining tools do not take entire advantage of these architectures. This paper exploits an approach to improve and enhance the performance of Weka, a popular data mining tool, through parallelization on GPU machines. From the profiling of Weka object-oriented code, we chose to parallelize a multiplication matrix method using state-of-the-art tools. The implementation was merged into Weka so that we could analyze the impact of parallel execution on its performance. The results show a significant speedup on the target parallel architectures, compared to the sequential Weka code. In this first experiment parallelized the Multiplication method, adapting the work method to take advantage of CPUs or GPUs according to the size of the matrices. As a result, can observed speedup levels of at least 47 percentage, drastically decreasing the time the algorithm consumes to handle a dataset.

Keywords: GPU, GUI, CUDA.

1. INTRODUCTION

Different alternatives can be considered in order to improve data analysis performance. While massively parallel data management systems have been used to scale the data management capacity, analytics and data integration analytics have increasingly become a lot, and can only get worse. In the open source scenario, we find, at one side, some recently appeared tools like Mahout and Vowpall Wabbitt which can perform big data analytics in large computer clusters. On the other hand, we have desktop based tools like R, Weka and Rapid Miner, which are usually employed in smaller yet important problems. Considering the computational complexity of some data mining algorithms, data analytics may take hours to complete. While some users will rely on cloud based solutions, heterogeneous environments based on GPU architectures appear as a valuable solution to improve performance with significant cost saving. Using a popular clustering algorithm, K-Means the large scale analytics accelerated GPU tool. As pointed out by Schad, using these environments, data can be stored and analyzed locally without requiring a specialized structure. This represents an interesting asset when considering access control and privacy problem that may concern analyzed data. GPU architectures represent then a complementary solution face to cloud environments. Nevertheless, several authors underline the complexity of using such architectures that often require significant expertise on GPU programming and related technologies. Such complexity may become an important limitation due to the growing popularity of data analytics for non-technical users. Most approaches to parallel data mining focus on distributed execution of multiple experiments, or specific parallel algorithms that are not integrated into popular data mining tools. In order to demonstrate the interest of GPU architectures for improving data analysis performance on desktop data mining tools, we propose, in this paper, an exploratory work in which we adapt a popular tool named Weka (Waikato

*Corresponding Author

Environment for Knowledge Analysis) to a GPU use. Weka is a popular open source tool for data mining, and it comprises a collection of data mining and machine learning algorithms packed with together in a feature rich toolbox, which includes pre-processing and analysis tools. Weka provides users with a Graphical User Interface and a Java-based Application Programming Interface. It implements algorithms for regression, classification, clustering, association rule mining and attribute selection, visualization. With all these features, Weka is commonly used in business research and education industries. Parallel and distributed computing was not a concern about the design of Weka, but it has been addressed by a few Weka projects, as for example Weka Parallel and Grid Weka. Furthermore, the development branch of Weka includes a package called Weka Server, which allows multiple servlet driven process instances that can run multiple tasks in parallel. Even so, parallel processing with gpus is still underexploited by Weka. All of Weka's techniques are predicated on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric or nominal attributes, but some other attribute types are also supported). In this paper, we explore further opportunities for parallel execution. Our main goal is to improve the performance enhancement and reduce the end user response time of WEKA on a single machine, taking advantage of GPU accelerator.

2. WEKA: MACHINE LEARNING TOOL

Waikato Environment for Knowledge Analysis. Its a data mining/machine learning tool developed by Department of Computer Science, University of Waikato, New Zealand. Weka is also a bird found only on the islands of New Zealand. Weka is a popular open source tool for desktop data mining. Containing a collection of data mining and machine learning algorithm packed together in a feature-rich toolbox. Weka is widely used in business, research and education. Weka provides users with a Graphical User Interface (GUI) and Java-based API. There are three GUI available

The Explorer (Exploratory data analysis)

The Experimenter (Experimental Environment)

The Knowledge Flow (New process model inspired interface)

2.1 Features of WEKA GUI

1. 49 data preprocessing tools
2. 76 classification/regression algorithms
3. 8 clustering algorithms
4. 3 algorithms for finding association rules
5. 15 attribute/subset evaluators + 10 search algorithms for feature selection

The core package contains classes that are accessed from almost every other class in weka. The most important classes in it are attribute, instance and instances. An object of class attribute represents an attribute it contains the attribute name, its type and in case of nominal attributes its possible values. An object of class instance contains the attribute values of particular instance, and an object oriented class Instances contains an ordered set of instances in other words, a dataset.

Fig 1: Package hierarchy

3. PROFILING WEKA

Profilers are tools that allow the gathering and analysis of execution characteristics of a program. One of the main uses of profilers is to evaluate the different software elements and to detect which parts of the code are more computing intensive or introduce slowdowns. This can be helpful as a starting point to software parallelization. If we knew the correct profile for a program run, we could evaluate the profiler with respect to this correct profile.

Unfortunately, there is no correct profile most of the time and thus we cannot definitively determine if a profiler is producing correct results. For this reason, we relax the notion of correctness into actionable. By saying that a profile is actionable we mean that we do not know if the profile is correct; however, acting on the profile yields the expected outcome. For example, optimizing the hot methods identified by the profile will yield a measurable benefit. Thus, unlike correctness which is an absolute characterization (a profile is either correct or incorrect), actionable is necessarily a fuzzy characterization .To identify the most computing intensive procedures on Weka hotspots we focused on a subset of algorithms that are common to several Weka applications. We selected the M5P algorithm, which is responsible for the creation of decision trees that will be used in regression models. This algorithm was suggested by experts in data mining, which had used the M5P in a previous job. Several profilers for Java language are available, with different strengths and strategies to collect information about a program. For this reason, the results of different profilers vary. It also worth note that the use of profilers is an intrusive approach that slows down considerably the execution of a program, diluting the impact of external factors like I/O and network accesses. Having that in mind, this work combined the analysis from different profiling tools, merging the results and focusing on the methods that most appeared. The tools we used are VisualVM, JProfilers, JProb as well as Java own integrated profiler (jprof).As expected, the performance and results from different profilers varied a lot. Indeed, VisualVM was exceptionally intrusive and slow, preventing the collection of any useful data. JProfiler had a better behaviour than VisualVM, showing a lower overhead and allowing the detection of some hotspots as illustrated in Fig1. JProb had the best behaviour concerning the profiling overhead, and it highlighted several hotspots pointed by JProfilers. Finally, the Java profiler showed useful, even though it does not show detailed information.

In addition to the overhead caused by the profiling applications, another important factor that must be considered relates to the choice of M5P algorithm parameters. Indeed, this algorithm allows different parameter combinations when analyzing the dataset, and we noticed differences between the profiling results when these parameters were changed. Therefore, the method that caused most impact in the performance was chosen by crossing the results of the different parameters with the output from different profilers, resulting in the list illustrated below fig 2.

Fig 2: Hot spot identified in weka

Most of these methods are well-known to the parallel computing community and can be ported to GPUs. In this work we decided to evaluate the impact of GPUs through the parallelization of a single method, the Matrix Multiplication as this method was presents a high number of calls with a non-negligible impact on the time they consume.

4. GPU IMPLEMENTATION

Since Weka is written in Java, and since Java does not support GPU devices directly, we had to find ways to do like. The Java policy of Write once, runs everywhere makes difficulty to implement such feature due to several difficulties such as hardware detection and byte code adaptation at runtime. Therefore, the simplest path is to use Javas native interface to call CUDA (Compute Unified Device Architecture) functions in C. In java we are using JCUDA. JCuda is a stable software suite that provides an extension to CUDA Basic Linear Algebra Subroutines (CUBLAS), called JCublas, which we found useful to adapt the matrix multiplication (sequential) to a GPU-enabled one. Please note that not all matrix multiplications are fit for GPU acceleration. Hence, found that M5P uses mostly rectangular matrices of different sizes, and sometimes these matrices are too small to benefit from GPUs due to the overhead of data transfer to and from the GPU.

5. EXPERIMENTS

To conduct these experiments, executed the sequential and the GPU improved version of Weka on three different machines. These machines contain different processors and NVIDIA GPUs, which allow us to infer on the contribution of both CPUs and GPUs to the overall performance of Weka. In order to evaluate the impact of GPUs on Weka, experiments considered three different approaches:

Sequential execution

GPU code only

A mixed code that selects best approach

The GPU implementation improved considerably the M5P performance on LSC4, as can be seen in Fig.3 As less the number of attributes per leaf (-M), more of them are needed, augmenting the height of the tree. When the tree is higher, the matrix multiplication algorithm is called more times and the speedup is more significant. On the other hand, when the tree is smaller, the accelerated matrix multiplication method was not called so many times, and its impact in the overall performance enhancement was less significant but yet considerable. The mixed approach (GPU-CPU) showed itself the fastest approach, as the use of GPU-only brings an unnecessary overhead when the matrices are too small.

Fig 3: M5Ps performance in LSC4 node using different approaches

6. RESULTS

The femiliarization of GPUs represents new opportunities for software parallelization and performance accessible to the final user on its own desktop. Data mining is one of the main fields that can benefit from these advances. In this paper, we presented the results of a first experience to enhance the performance of Weka data mining software, a well-known data mining tool. Thanks to the use of Java profilers, we identified a set of operations that are time-consuming and that can easily be adapted to GPUs. In this first experiment we parallelized the Multiplication method, adapting the work method to take advantage of CPUs or GPUs according to the size of the matrices. As a result, we observed speedup of at least 47 percentages, drastically decreasing the time the algorithm consumes to handle a dataset. Future works shall continue the parallelization of other Weka algorithms on GPUs, but also improve the efficiency of CPUs on Weka, which for the moment is not able to take all the advantages of multi-core architectures. .

REFERENCES

[1] Tiago Augusto Engel a , Andrea Schwertner, Manuele Kirsch-Pinheiro. Performance Improvement of Data Mining in Weka through GPU Acceleration, 5th International Conference on Ambient Systems, Networks and Technologies(ANT-2014)

[2] Schadt,E,Linderman,M.D.,Sorenson,J.,Lee,L.,Nolan,G.P..Computational Solutions to large-scale data management and analysis. Nature reviews genetics 2010;11(9):647657.

[3] Wu, R., Zhang, B., Hsu, M.. Gpu-accelerated large scale analytics. Tech. Rep. HPL-2009- 38; HP Labs; 2009.

[4] Ma, W., Agrawal, G.. Auto-gc: automatic translation of data mining applications to gpu clusters. In: 24th IEEE International Symposium on Parallel and Distributed Processing - Workshop Proceedings. IEEE Computer Society; 2010, p. 1 to 8.

[5] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P.,Witten, I.H.. The weka data mining software: an update. SIGKDD Explor Newsl 2009;11(1):10 to 18.

[6] Perez, M.S., S anchez, A., Herrero, P., Robles, V., Pe na, J.M.. Adapting the weka data mining toolkit to a grid based environment. In: Advances inWeb Intelligence (AWIC); vol. 3528 of Lecture Notes in Computer Science. ISSN: 0302-9743. Lodz, Polonia: Springer; 2005, p. 492 to 497.

[7] Ghoting, A., Kambadur, P., Pednault, E., Kannan, R.. Nimble: A toolkit for the implementation of parallel data mining and machine learning algorithms on mapreduce. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; KDD 11. New York, NY, USA: ACM. ISBN 978-1-4503-0813-7; 2011, p.334 to 342.

[8] Kumar , p , Ozisikyilmaz, B, Liao, W K Memik,G.choudhary A High performance data mining using r on hetero geneous platform In: Parallel and distributed processing workshop and phd form(IPDPSW),2011 IEEE International symposium on 2011 p 1720 to 1729

[9]Talia, D., Trunfio, P., Verta, O.. Weka4ws: a wsrfenabled weka toolkit for distributed data mining on grids. In: Proc. of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2005. Springer-Verlag; 2005, p. 309 to 320