Online Face Recognition Using Big Data From Video Stream By Bandit Segment

Shruthi Hiregoudar[1], Dr. Revathi V[2]
[1, 2] Dept. of Computer Science & Engineering
[1, 2] Dayananda Sagar University, Bangalore

Abstract- *The earlier scheduling techniques did not have any resource allocation to the streaming task at run-time. An assumption was made that the system knew how much resource has to be allocated to each task in order to achieve the desired throughput. In traditional scheduling method, the problem was formalized as a Stochastic Shortest Path (SSP) problem, and a reinforcement learning algorithm was used to learn the environment dynamics. The SSP solution cannot be applied to multi-stream. To overcome these drawbacks Staged Multi-armed Bandits (S-MAB) scheduling methodology is proposed, which uses the novel class of online adaptive learning techniques. This scheduler can assign particular processing method to multi-streams at a time based on the requirement. Moreover, also allocate resources over time to maximize the performance at run-time, without having to access any offline information.*

The proposed scheduler has to be applied on a face detection streaming application, which does not use any offline information. The scheduler can achieve similar performance when compared to an optimal semi-online solution that has full knowledge of the input stream. The scheduler also selects actions in the current stage based on the feedbacks and actions of the past stages.

Keywords- Scheduling, Stochastic Shortest Path (SSP), Staged Multi-Armed Bandits (S-MAB), Multi-Stream Processing

I. INTRODUCTION

Several domains make use of Big Data streaming applications such as social media analysis, financial analysis, video annotation, surveillance and medical services. However, the parallel computation of the data stream was a challenge. To increase parallel computation requirement and an extremely changeable random input data stream which influences directly the application difficulties and final Quality of Service QoS (e.g., throughput and output quality) [6], these applications are classified with rigid delay constraints. For example, stream mining applications [5], are used to classify an excessive input of varying data stream and are used to model a series of stages of classifiers and features-extraction jobs. Multiple types of classifiers are applied to the variety of actively changing data that are gathered from various diverse sources to uncover hidden patterns or extract knowledge necessary for prediction and actionable intelligence applications. To adjust the heterogeneous type of data, every stage may merge with various type of quality levels or classifiers, and a processing method is chosen at run-time on the estimated type of data. The complication of each work in each stage of the chain may vary at run-time with respect to the kind of processed input data which is unknown by the application.

Numerous hardware and software solutions have been put forward to handle the growing complications and computation requirement of modern streaming applications. At the hardware layer, various many-core architectures have been established to make the parallelism high and to assist the streaming application model. At the software layer, approaches depending on load-shedding method have been suggested to reduce the job by choosing the quantity of data that will be processed while other methods control the processing style of the data streams to adjust to the given allocated resources. However, most of the state-of-the-art solutions do not handle multiple streams simultaneously. Moreover, even in the single stream instance, without the assistance of a proper online smart scheduler, these many-core platforms are not able to efficiently handle real-time requirements and characteristics of Big-Data streaming application which are actively changing at an instant of time. In fact, existing online scheduling methods have very less attention to the live attributes of the data streams, which may undergo concept drift [2] and thus need continuous modification. Methods that depend on offline information cannot adjust to these concept drifts online.

At last, energy utilization in many-core architectures is becoming a major concern as the cost of powering these types of platforms is significantly increasing [8]. Software techniques described in the previous paragraph adapt the complication of stream mining applications at run-time. However, such workload reduction results are usually executing at the application layer which is often unaware of the system architecture, available system resources or available power handling features. Therefore, by integrating these software methods with energy-conserving characteristics such as Dynamic Power Management (DPM) to switch on and off cores, the energy usage can be reduced without having an influence on the quality of service of the application. In fact, by allocating the proper quantity of resources to each task, only required cores is activated. Moreover, the average time between different application stages can be utilized with DPM when it is identified. Therefore, it is necessary that the operating system layer integrates approaches from both application and hardware layers to improve the QoS while reducing the energy consumption.

To solve this kind of issues, a new methodology and related algorithms for online learning and energy-efficient scheduling of Big-Data streaming applications with multi-streams on many-core systems with resources constraint are proposed.

The proposed scheduler is applied to a multi-stage face detection streaming application in a dynamically varying environment. Moreover, without handling any offline information, it can attain related performance compared to an optimal semi-online solution. It has full knowledge of the input stream where the differences in throughput, observed quality, resource usage and energy efficiency are below 1%, 0.3%, 0.2% and 4% respectively. Comparison of results has to be made with scheduling solution [1] with online learning and concept drift detection. The scheduler operates better than the solution proposed in [1] terms of observed quality, obtained throughput, memory usage, and complexity.

II. MOTIVATION

In the traditional method, Stochastic Shortest Path (SSP) is applied on a single stream, and it may not give a clear and systematic way to choose the tasks of each stage in exploration mode. The multi-stream model was not handled using SSP. For a given input size the core allocation was not done to accomplish the desired throughput. Priority is not given for the task selection. To overcome those kinds of problem, the Staged multi-armed bandit (S-MAB) scheduler and Haar Cascade training algorithm are used in proposed work.

III. RELATED WORK

Our approach targets a specific type of application where the QoS depends on both the throughput and the quality observed for each single task in the application with a dynamic Big-Data stream under constrained resources. Therefore, we only consider techniques that are recommended to adjust Big-Data streaming applications to resource constraints.

The first set of techniques depends on load-shedding [11] [10], where designed algorithms determine how much quantity of data should be discarded and when, where, what, given the desired QoS requirements and the present resources. In [11], the effect of load shedding is known before, and the load shedder is separated from the scheduler imagine that an external scheduler will take care of the assignment of freed resources. In [10], a load shedding scheme ensures that dropped load has the negligible effect on the advantages of mining and dynamically learns a Markov model to predict feature values of hidden data. Instead of determining what quantity of data to process, as seen in load shedding, the next set of techniques [9] [7] [4] [3] [5] decide how the present data should be processed when the underlying resource allocation is given. In these activities, individual tasks operate at a different performance level when the resources are assigned to them. They imagine a fixed model complexity for each classifier and the differentiation of the output quality is known a-prior. The problem was framed as a network optimization problem and solved with sequential quadratic programming. These solutions assume static environment while, in reality, data streams are dynamic. Therefore, they may undergo a concept drift that needs a constant online adjustment of the quantity of allocated resources to the individual task and the output quality to increase the QoS, especially when the resources are constrained. In [2], a study has been printed recently, which classifies most of the existing concept drift approaches. None of these approaches have been suggested for scheduling Big-Data streaming applications and resource management problems. Recently, in [1], the authors represent the scheduling problem as a Stochastic Shortest Path problem and recommend a reinforcement learning algorithm to understand the environment dynamics to resolve this issue even with the existence of concept drift. However, the allotment of the computing resources to each streaming task was not realized by the algorithm. Instead, it assumes that the system knew the number of resources to pass on to each work to accomplish the desired throughput. Moreover, they will not give an organized way for the task selection.

To outline, these two sets of results are usually executed at the application layer and are unsure of the system constraints and capabilities. Instead, our online learning solution is executed at the operating system level, and it is responsible for resource allocation and processing method selection for each available stream.

IV. LITERATURE REVIEW

"Configuring trees of classifiers in distributed multimedia stream mining systems" was proposed by B. Foo, et al. [3], in which the data is dynamically filtered and processed. Different aspects of data content are recognized. Therefore dependability on a group of cascaded statistical classifiers is required by multimedia stream mining applications. To configure such cascaded classifier topologies, specifically binary classifier trees, in resource-constrained, distributed stream mining systems a novel approach is introduced. By mutually taking into account the misclassification cost of each end-to-end class of concern in the tree, the resource constraints for each classifier and the confidence level of each data object that is classified classifiers are configured with best operating points instead of conventional load shedding. Dynamically intelligent load shedding and data replication are allowed depending on given resources. The algorithm is evaluated on a sports video hypothesis recognition application and identifies immense cost savings over load shedding alone. To reconfigure every classifier in the tree by itself various distributed algorithms are proposed depending on restricted information exchange. For each algorithm, an associated trade-off is analysed between convergence time, information overhead, and the cost efficiency of results achieved by each classifier.

"A distributed approach for optimizing cascaded classifier topologies in real-time stream mining systems" was proposed by B. Foo, et al., [4]. For arranging classifiers in a real-time, informationally-distributed stream mining system, distributed optimization techniques are discussed. Stream mining systems need to bear with overload due to the huge volume of streaming data, because of which the real-time applications will suffer bad performance and unbearable processing delay. Feature value of data arriving at classifier is the classification performance achieved by a group of classifiers, and the end-to-end processing delay is impacted by changing the filtering process at one classifier when the whole system of classifiers is considered for optimization, which becomes a difficult task. To address this problem, three main contributions are made. 1) To improve performance, the ordering of classifier filtering elements is optimized and more general topologies are used, such as parallelization of some filtering elements. 2) By noticing, approximating, and interchanging variables between the similar classifiers deployed across the system, a low-complexity framework is introduced for approximating the system utility. 3) depending upon their convergence properties, optimality, information exchange overhead, and rate of adaptation to non-stationary data sources, distributed algorithms are provided to configure the system and analyse the algorithms.

"Adaptive topologic optimization for large-scale stream mining" was proposed by R. Ducasse, et al. [5]. In big-volume data stream, the real-time classification and recognition of particular features are critical for a huge number of applications, together with large-scale multimedia analysis, processing, and retrieval. Using a group of binary classifiers the required data is filtered that are established on distributed resource-constrained base.

To trade-off accuracy of feature identification with filtering delay, the focus was made on selecting optimal topology (series) of classifiers and present algorithms for classifier ordering and configuration. Data characteristics, system resource constraints, performance and complexity characteristics of each classifier are considered for order selection. The centralized Greedy algorithm is used for ordering the series of classifiers that leads to a close-to-optimal utility, but the timely adaptation to system dynamics is not allowed. To satisfy these requirements, a decentralized approach was proposed, in which classifiers can make decisions autonomously with limited message exchange.

F. Fu, et al., proposed "Configuring competing classifier chains in distributed stream mining systems" [7], in which System and algorithmic researchers are attracted towards a network of classifiers as they provide improved accuracy over single model classifiers. It can be distributed over a network of servers for improved scalability and can be adjustable to available system resources. The algorithms are designed for the optimal configuration of competing series of classifiers which are distributed across a resource constrained stream mining system.

For the speaker verification application, classifier cascade is used alternative to single classifier under resource constraints. Among several classifier chains competing for system resources, centralized and distributed algorithms are designed to provide efficient and fair resource allocation. Nash Bargaining Solution (NBS) from game theory is used to ensure this. Under both balanced and unbalanced rate and resource constraints, the accomplishment of centralized and distributed algorithms on an application is analysed with competing classifier chains. The result that shows two algorithms converge to the same optimal solution is obtained, with no duality gap between them. However, the rigorous proof of the global convergence and zero duality gap is not provided.

V. COMPARATIVE ANALYSIS

"Configuring trees of classifiers in distributed multimedia stream mining systems" was proposed by B. Foo, et al. [3], In which the dynamic and domain-specific feature extraction at each classifier is not provided.

"A distributed approach for optimizing cascaded classifier topologies in real-time stream mining systems" was proposed by B. Foo, et al., [4], in which instantiation of multiple classifier elements on each node with respect to shared resource constraints (i.e., time-sharing on a CPU) was not possible.

"Adaptive topologic optimization for large-scale stream mining" was proposed by R. Ducasse, et al. [5]. It cannot handle uncertainty in the information about classifier operating characteristics, data statistics, and resource variability.

F. Fu, et al., proposed "Configuring competing classifier chains in distributed stream mining systems" [7], in which rigorous proof of the global convergence and zero duality gap is not provided. Also, duality gap between the centralized and distributed algorithms (with overall utility difference of 12%) is observed, for the scenarios with classifier sharing across chains.

The proposed paper overcomes all the above issues by using the Staged Multi-Armed Bandits (SMAB). It introduces online learning and energy-efficient scheduling of Big-Data streaming applications with numerous streams on many core systems with resource constraints.

Table 1. Comparison between S-MAB and SSP Throughput

	S-MAB	SSP
Rounds 1-300	65%	30%
Rounds 301-600	55%	25%
Rounds 601-900	70%	30%

The above Table 1 shows the comparison between S-MAB and SSP Throughput at different stages for specific rounds of computation.

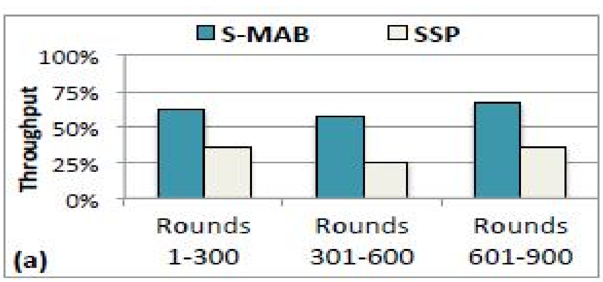

Figure 1. Performance comparison between S-MAB and SSP for the obtained global throughput.

The above Fig. 1 shows the performance comparison between S-MAB and SSP for the obtained global throughput at different rounds of computation.

Table 2. Comparison between S-MAB and SSP Average Quality

	S-MAB	SSP
Rounds 1-300	100%	85%
Rounds 301-600	100%	65%
Rounds 601-900	100%	70%

The above Table 2 shows the comparison between S-MAB and SSP average quality at different stages for specific rounds of computation.

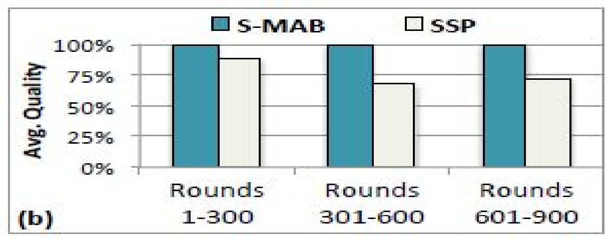

Figure 2. Performance comparison between S-MAB and SSP for the observed average quality.

The above Fig. 2 shows the performance comparison between S-MAB and SSP for the observed average quality at different rounds of computation.

VI. CONCLUSION

A new standard and well-planned methodology and related algorithms for online learning and energy-efficient scheduling of Big-Data streaming applications with numerous streams on many core systems with resource constraints are proposed. The main offerings of this task are as follows: (1) Formalized the issue of multi-streams scheduling as a staged decision issue in which the performance obtained for several resource allotments is not known before but known over time. (2) The scheduler can determine which processing method to set to each stream and how to assign resources over time to make the performance high on the fly, at runtime, without having any offline information. (3) Unlike other online learning techniques such as standard multi-armed bandits and reinforcement learning, in our formulation, the outcome of every scheduling action depends on a series of prior scheduling conclusions and feedbacks that are taken at a specific stage (window) of time.

REFERENCES

[1] K. Kanoun, et al., “Big-Data streaming applications scheduling with online learning and concept drift detection” in Proc. DATE, 2015.

[2] J. Gama, et al., “A survey on concept drift adaptation,” ACM Computing Surveys, 2014.

[3] B. Foo, et al., “Configuring trees of classifiers in distributed multimedia stream mining systems,” in IEEE TCSVT, vol. 21, no. 3, pp. 245–258, March 2011.

[4] B. Foo, et al., "A distributed approach for optimizing cascaded classifier topologies in real-time stream mining systems," in IEEE TIP, vol. 19, no. 11, pp. 3035–3048, November 2010.

[5] R. Ducasse, et al., “Adaptive topologic optimization for large-scale stream mining,” in IEEE JSTSP, vol. 4, no. 3, pp. 620–636, June 2010.

[6] C. Ballard, et al., "IBM InfoSphere Streams. Harnessing data in motion," IBM Redbooks 2010.

[7] F. Fu, et al., “Configuring competing classifier chains in distributed stream mining systems,” in IEEE JSTSP, vol. 1, no. 4, pp. 548–563, December 2007.

[8] J. G. Koomey, et al., “Estimating total power consumption by servers in the U.S. and the world,” Stanford Univ. Press, 2007.

[9] D.S. Turaga, et al., “Resource management for networked classifiers in distributed stream mining systems,” in Proc. ICDM, 2006.

[10] Y. Chi, et al., “Loadstar: Load shedding in data stream mining,” in Proc. VLDB, 2005.

[11] N. Tatbul, et al., “Load shedding in a data stream manager,” in Proc. VLDB, 2003.