AN EFFECTIVE SPARK BASED RESOURCE MANAGEMENT FOR IMPLEMENTATION OF FP-GROWTH ALGORITHM IN CLOUD ENVIRONMENT

[1]A.Senthilkumar, [2]Dr.D.Hariprasad
[1]Research Scholar, Sri Ramakrishna College of Arts and Science,
Coimbatore-641 006
[2]Professor and Head, Department of Computer Application,
Sri Ramakrishna College of Arts and Science,
Coimbatore-641 006

Abstract

The information generated from different sources such as mobile devices, sensors, web cameras in day to day life is growing exponentially and is processed in big data. These processed data have become high important for all the major domains, such as research, business and industry. One of the big data processing platform is the Apache Spark which can handle both batch processing and real time streaming data. Cloud computing which can provide required resources are used to meet the real time processing requirements of streaming applications. In a virtualized cloud environment, where multiple big data applications are deployed, the performance interference can also affect the performance of the streaming tasks resulting in the performance degradation of the jobs. Association Rule Mining is the algorithm used to find the strongly related patterns between itemsets. FP-Growth algorithm is the widely used Association Rule mining algorithm. In this paper, the parallel FP-Growth algorithm is implemented in Spark Framework. The execution time is reduced and the cost is optimized by effective utilization of Spark resources with heterogeneous resource allocation.

Keywords: Bigdata, Apache Spark, Cloud Computing, Association Rule Mining, Fp-Growth

1. Introduction

Association Rule Mining is a machine learning algorithm which finds pattern or relations between itemsets[1]. FP Growth is one of the efficient association rule mining algorithm which uses divide and conquer concept. When compared to other association rule mining algorithms FP-Growth is faster and it is linearly scalable. Some of the applications of Association Rule Mining algorithm are Market Basket Analysis, Global Information System. Medical Diagonosis, Collaborative filtering, Threat/Crime detection and so on.

There are enormous data generated from medical applications, social medial data, online shopping etc. To find the pattern from this large dataset using traditional association rule mining algorithm is a tedious task and to improve the performance of this algorithm Bigdata framework is used. The In memory capability of Spark platform makes the association rule mining algorithm more efficient and fast for the large datasets. Cloud computing which can provide required resources are used to meet the real time processing requirements of streaming applications. Hence the proposed system uses the Cloud resources with Spark Framework to execute the Association rule mining algorithm. Spark is used in the proposed system as it has faster processing in memory capability when compared to Hadoop.

Big data deals with large amount of structured, unstructured and semi structured data[2][3][4]. Apache Spark is one of the big data processing platforms. It is a fast and general MapReduce like engine for large scale data processing in a local or a cloud cluster which can be used for both batch processing and real time processing [5-7]. Spark is designed to cover a wide range of workloads such as batch applications, iterative algorithms, interactive queries and streaming interactive queries and stream processing. The main feature of Spark is its in-memory cluster computing that increases the processing speed of an application. The apache Spark framework is shown in Fig1.

Spark application run as a single node or independent set of processes on a cluster in the driver program coordinated by the SparkContent Object. SparkContext connects to cluster manager which can be spark standalone cluster, Mesos or YARN and allocates resources to different applications. Spark application consists of a driver program and executors on the cluster. The driver program creates the SparkContext. Worker node is any node that run application in a cluster. Executors are launched when the spark application begins and runs for the entire lifetime of an application Executors are worker nodes which executes tasks and each application has its own executors.

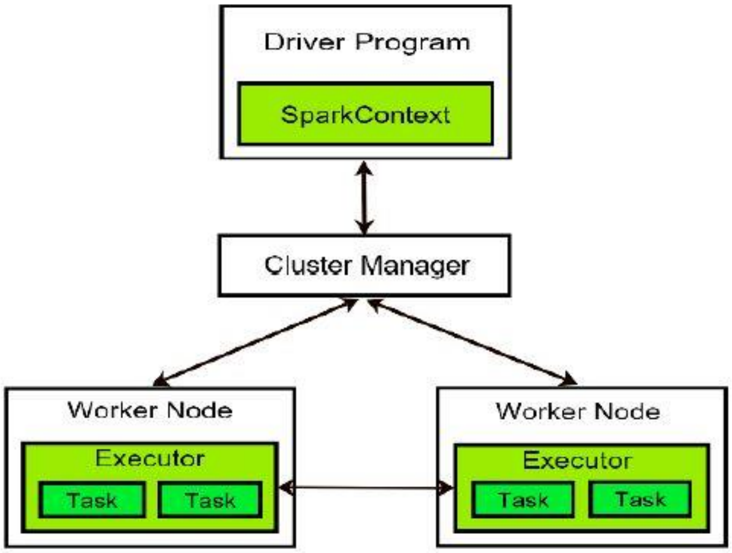

Fig 1: Apache Spark Architecture

Cloud computing is an accelerating technology which is a form of distributed computing which can provide services such as Infrastructure as a Service(IaaS), Platform as a Service(PaaS) and Software as a Service(SaaS)[8]. Cloud has a shared resource pool which can be used on demand. Virtualization, which is the key concept of cloud enables creation of multiple version of something such as a operating system, server or any other network resources. Virtual Machines are used by the cloud providers to provide CPU memory, storage and the complete infrastructure. The demand to these resources based on the application and hence dynamic allocation of resources for different types of an application is the major issue in cloud.

Two major Bigdata tools are Hadoop and Spark, Spark can execute jobs upto 100 times faster than hadoop with same amount of resources. Both Hadoop and Spark have linear scalability property, with more resources it can run the job faster. Cloud offers virtually unlimited resources on demand. These two technologies are integrated in our solution for running bigdata workloads in shortest amount of time in least amount of resources.

Rest of the paper is organized as follows. Section 2 describes the literature survey where existing work related to the paper is discussed. The proposed methodology in section 3 describes the method used in Spark for effective resource utilization in cloud infrastructure for FP Growth Association rule mining algorithm. The experimental evaluation which shows the performance of the model is shown in Section 4 followed by conclusion.

2. Literature Review

There are enormous number of research efforts that addresses the issues of executing association rule mining in parallel and distributed environment. Most of the work focused on improving the execution is association rule mining algorithm is distributed environment.

A rare itemsets mining algorithm based on RP-Tree and Spark framework is proposed in [9]. The dataset is arranged vertically based on the identifier of the transaction and thus it solves the problem of scanning the entire dataset. To make the task simpler the dataset is divided into vertical dataset and rare vertical dataset. Frequent pattern tree is generated for the rare item dataset. The support for the dataset is calculated by scanning the vertical datasets.

Frequent Itemset mining is performed on the entire dataset in an Apriori algorithm and when the data is very large it becomes more tedious. Big data SPark Framework is used by the authors in [10] to optimize the execution time. They use hybrid frequent itemset mining and pattern is determined using vertical and horizontal structure of the database. It scans the horizontal transaction data and produces k-frequent itemsets. Then the support of candidate items is computed by exploring vertical data on each node. It has k sets of iteration and produces k-frequent itemset.

Kollenstart et al[11] used genetic algorithm to predict the utilization of resources and it predicts the ratio of instance precessing tasks as well as ratios of the resources that are used by these instances. The predicted ratios are used as part of an adaptive which can itself reconfigure to maximize utilization. By using this approach the distributions for processing steps and workers can be can be estimated and balanced. These distributions are then used in the control loop to continuously monitor applications at runtime which can be adaptive and which matches the actual demand of the application. CPU intensive task is taken for their evaluation where the amount of data that has to be transferred is low compared to the computational time and hence data locality is considered here. By performing computations as close to the data as possible, the performance loss in transferring data can be reduced. However, when heterogeneous resources are used, the data has to be moved to the corresponding platform which is unavoidable.

According to Ruiz-Alvarez, Kim and Humphrey[12], determining the type of resources and number of resources required is a challenging task in cloud environment. In their method they predicted the requested computation in terms Integer Linear Programming to make a provisioning decision in a few milliseconds. Two importance performance metrics used in their model is the cost incurred to execute the job and the execution time of the job. Two types of cloud applications, MapReduce application and Monte Carlo simulations are used for performance evaluation. A significant advantage in their approach is that their solution has been proved optimal by their method and the set of the scheduling decisions based on their model are plotted on a time vs. cost graph that forms a Pareto efficient frontier which are faster or cheaper than any alternatives.

The authors in [13] proved that the conventional approached for frequent itemset mining using big data environment poses a significant challenges when there are limited computing capability and also when the memory space is limited. In their work, a matrix based pruning approach is used to improve the performance of the algorithm which reduces the size of the candidate set generated. They proved the result by implementing using Spark to further improve the efficiency of iterative computation.

3. Proposed Methodology

For a large scale task processing like Association rule mining, where it can be compute intensive jobs or data intensive jobs, utilization of resources plays an important role. On Large dataset, the existing FP Growth algorithm to compute frequency item set it takes more time to execute, which degrades the performance of the system. The job being processed for finding the pattern may require heterogeneous types of resources and it is a challenging issue to identify the resource utilization. By using cloud resources which enables the use of dynamic clusters of resources, where the size of the resources can also be altered dynamically. Cloud can provide efficient flexible resource provisioning, but the overall performance interferences for different types of application may affect the efficiency of effective resource utilization. Hence, the proposed work is to model cloud based effective resource management for Association rule mining applications using Apache Spark which can compute frequent item sets on massive data quickly.

An overview of the proposed methodology is listed below

- Heterogeneous resource management and Dynamic cluster provisioning for FP-Growth
- Cost optimization due to heterogeneous resource allocation

The overall framework of the Proposed methodology is shown in Fig 2. FP Growth based Association Rule mining applications are submitted to the Custom Resource Allocator, where the resources are for job execution are from cloud model. Spark master can dynamically allocate resources to the spark executor and once the jobs are completed, the resources are released to the cloud by the Custom Resource Allocator.

Cloud virtual machines are used to form clusters which can consist of master and worker nodes and in the proposed methodology the worker nodes are heterogeneous in nature which means that the nodes have different amount of CPU cores, memory, and disk storage which can be allocated dynamically based on Job nature. The worker nodes are terminated after the job completes to save cost as Cloud providers only charge for VM's running time. Heterogeneous Resource allocation provides optimal amount of resources for running FP-Growth algorithm which has CPU intensive workloads, hence we allocate CPU optimized or RAM optimized VMs as Worker nodes for running the corresponding jobs. This decreases both the cost and running time.

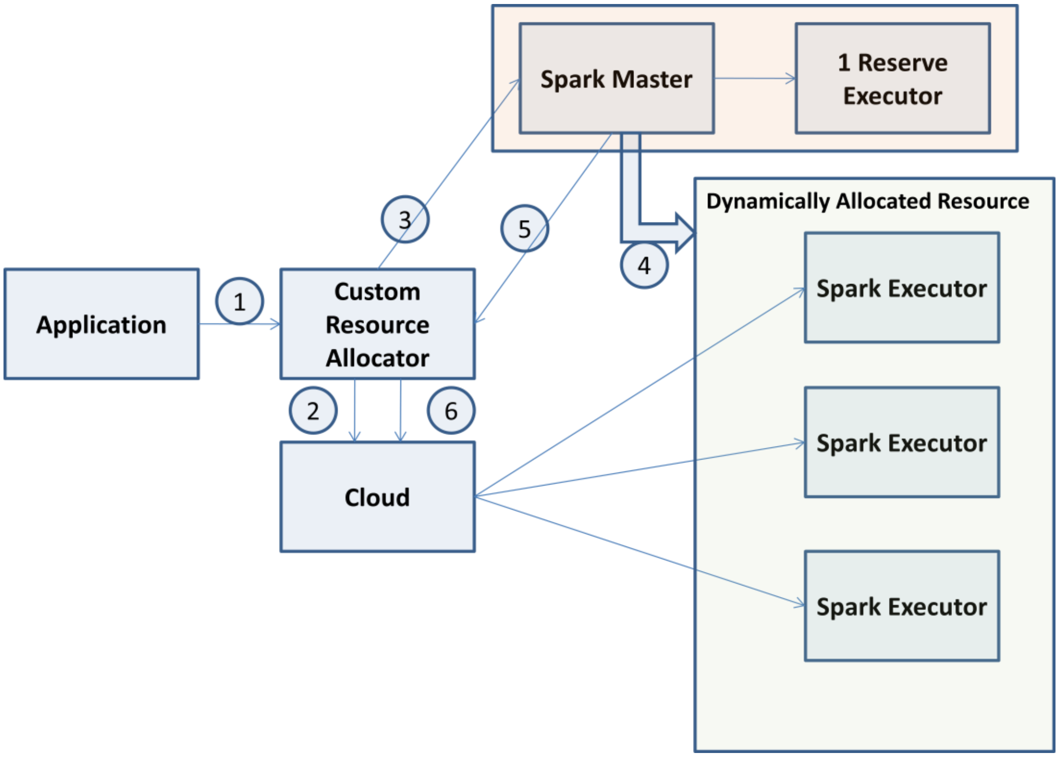

Fig 2: Framework of Proposed Methodology

Algorithm:

1. FP-Growth task is submitted to Custom Job Resource Calculator with input file location and nature of Job

2. The Custom Resource Allocator computes the number executers and the CPU & RAM resources necessary for each executer to run the job by based on a) Input Data size b) Job Nature

3. With the computed information of Number of Executers necessary, size of each container in terms of RAM & CPU, the Resource Allocator identifies the Number of Virtual Machines that needs to be created and VM Size

4. The Custom Resource Allocator creates the computed number of VMs in the cloud and add them to the Spark Cluster as slaves dynamically

5. The Customer Resource Allocator now submits the job on Spark Standalone dynamically allocated cluster with additional resources

6. After the Job completes, the additional new VMs created for execution of this job are terminated

The nature of jobs submitted on Spark cluster is Compute intensive job with large dataset. The jobs can be RAM intensive and CPU intensive correspondingly, hence running them on a Homogeneous resource allocation scheme will under utilize the resources and will give non-optimal performance in terms of Resource utilization and execution time. In the proposed work, both the nature of Jobs with various CPU and RAM ratios are executed and will interpret the effect of the CPU to RAM ratio on the execution time of the jobs submitted on spark.

.4. Experimental Evaluation

The proposed work is implemented using Scala Programming langauge with Ubuntu Server version 18.04 in a cloud environment with Apache Spark. FP-Growth algorithm, an iterative based applications with synthetic dataset is submitted to evaluate the performance of the proposed system.For FP-Growth, the different input workloads taken are 4.7 GB and 10 GB for 1-5 core CPU with Ram Size from 1 to 12 GB

Comparison of CPU and RAM resources with running count Fp-Growth based machine learning application is illustrated in FIg 3. CPU and RAM of various ratios and varying container sizes with the outcome of the job running time is plotted along with container size is illustrated in the graph. It can be inferred that the minimum container size one core with 1 GB RAM and 3 core with 6 GB RAM for FP-Growth applications.

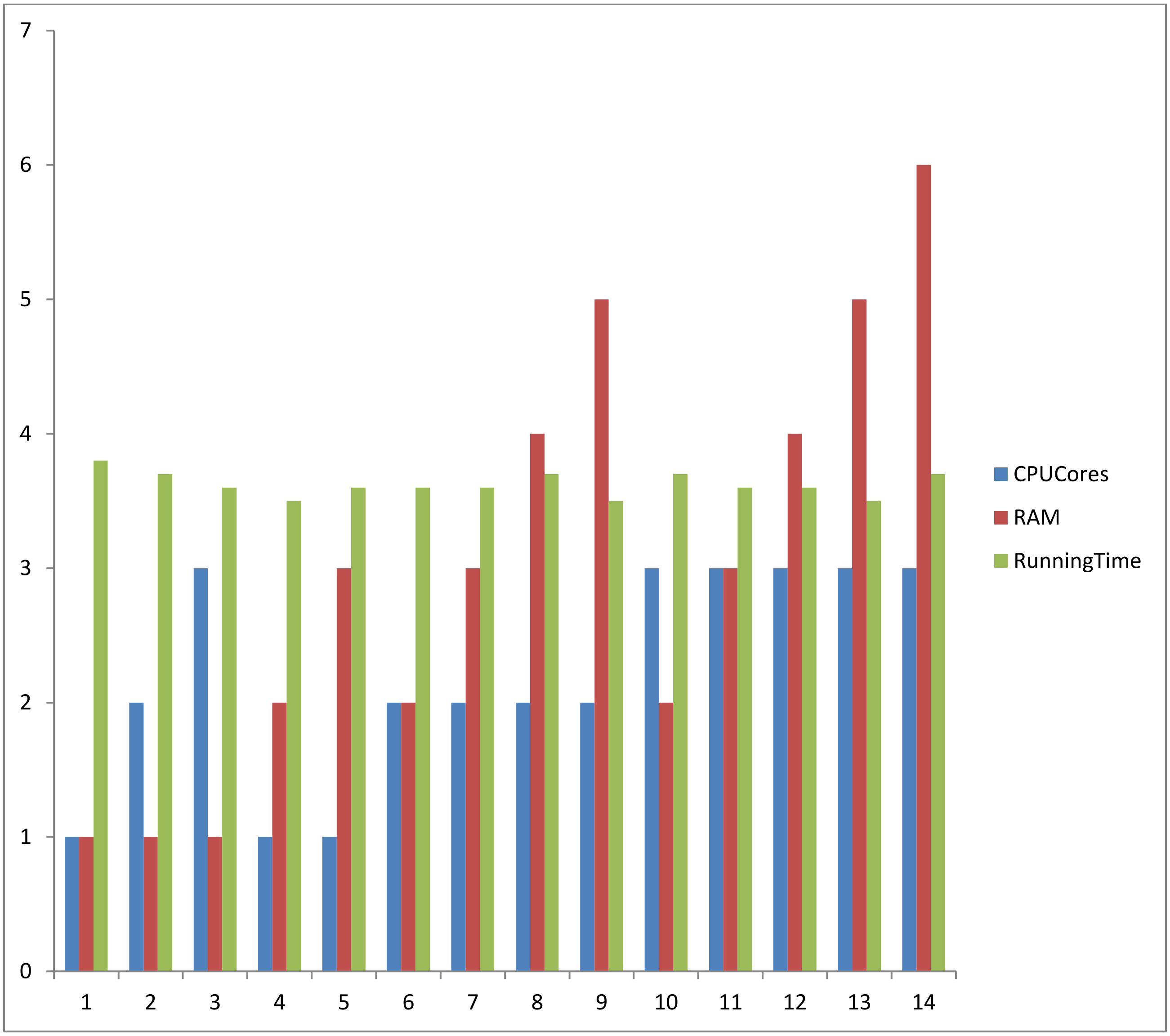

Fig 3: Comparison of CPU and RAM resources with Running Time for Machine Learning Application

As per the experimental results the observation is made on the impact of CPU and RAM resources with the effect on running time which is shown in Fig 4 and 5. The graph illustrates the optimal resource level and further allocation of resources beyond the optimal values either increases the execution time or remains the same resulting in a elbow curve as shown in the graph. From the graph we can identify per core 4 MB of data can be allocated for FP-Growth algorithm execution in Spark environment. The experimental results shows ideal CPU to RAM ratio for executing the algorithm is per core 750 MB of RAM can be allocated for optimal performance.

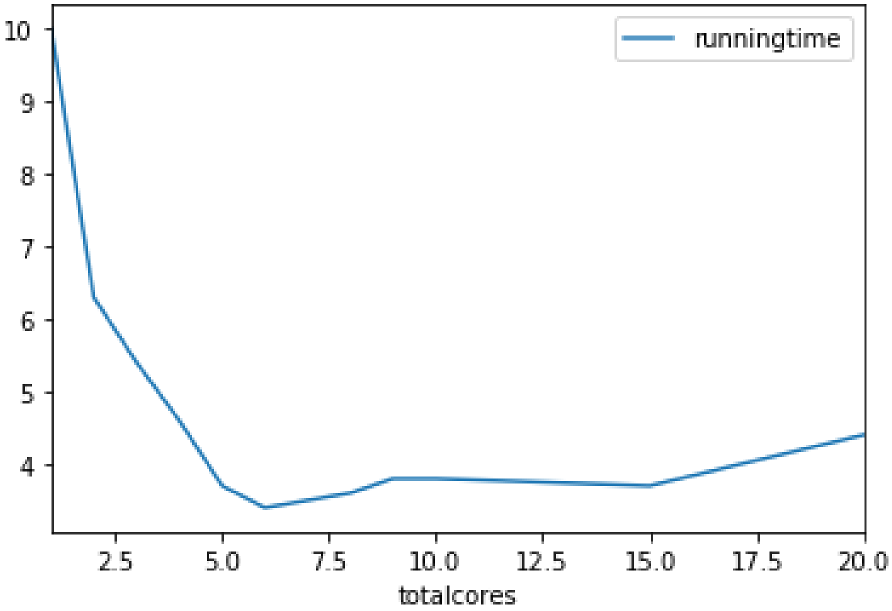

Fig 4 :Machine Learning Lbow Curve for 23 GB Data

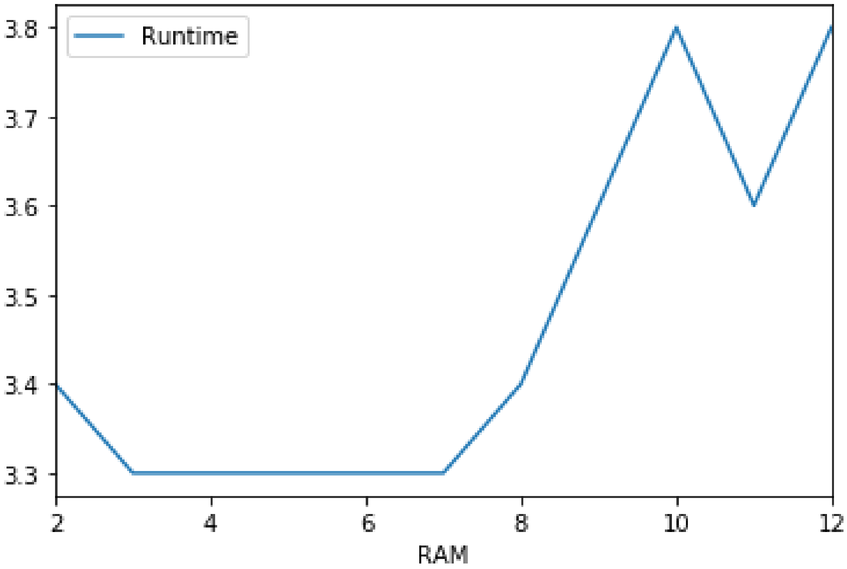

Fig 5 : Running Time Vs CPU & RAM Ratio FP-Growth

In order to reduce the execution time and resource consumption of the jobs, we consider Fp-Growth algorithm with the amount of data to be processed to compute the resources necessary to optimally run the job . The resource allocation scheme allocates 1:0.75 ratio for Jobs of nature with compute intensive work loads like Association Rule mining algorithm use cases which are the most common Machine Learning workloads. Hence we allocate more skewed type of containers with high CPU resources dedicated for these containers. Next the resource allocation scheme also identifies the amount of input data and based on the size of the data it decides the number of containers to be allocated, the optimal values are arrived based on our experimental evaluation. This calculation is based on the ratio of 1 CPU core for 4 MB of data for job nature for machine learning applications approximately.

Conclusion

The utilization of resources for a large scale task processing jobs which are compute intensive like FP-Growth plays an important role. By using cloud resources which enables the use of dynamic clusters of resources, the size of the resources can also be altered dynamically. Cloud can provide efficient flexible resource provisioning, but the overall performance interferences for different types of application may affect the efficiency of effective resource utilization.. In order to reduce the execution time and resource consumption of the jobs, the amount of data to be processed to compute the resources necessary to optimally run the job are considered. Hence more skewed type of containers are allocated with high CPU resources dedicated for these containers. Next the resource allocation scheme also identifies the amount of input data and based on the size of the data it decides the number of containers to be allocated, the optimal values are arrived based on the experimental evaluation. As a future work, dynamic prediction can be done to predict the most optimal resources using FP-Growth. So that after the jobs are run for a few times on the data, the resource allocation can auto tune the resource allocation according to the specific nature of features present in the data.

References

[1] Trupti A. Kumbhar, Santosh V. Chobe, "An Overview of Association Rule Mining Algorithms", International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 927-930,2014.

[2] A. Siddiqa, "A survey of big data management: Taxonomy and state-of-the-art", *J. Netw. Comput. Appl.*, vol. 71, pp. 151-166, Aug. 2016.

[3] Ranjeeth, S., Latchoumi, T. P., & Paul, P. V. (2020). A Survey on Predictive Models of Learning Analytics. Procedia Computer Science, 167, 37-46.

[4] Khezr Seyednima & Navimipour Nima, "MapReduce and Its Applications, Challenges, and Architecture: a Comprehensive Review and Directions for Future Research", Journal of Grid Computing. 15. 1-27. 10.1007/s10723-017-9408-0, 2017.

[5] Latchoumi, T. P., & Sunitha, R. (2010, September). Multi agent systems in distributed datawarehousing. In 2010 International Conference on Computer and Communication Technology (ICCCT) (pp. 442-447). IEEE.

[6] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica, “Apache spark: A unified engine for big data processing,” Communications of the ACM, 2016.

[7] Peter P. Nghiem. Best Trade-Off Point Method for Efficient Resource Provisioning in Spark. MDPI Algorithms 11(12), November 2018.

[8] Ranjeeth, S., Latchoumi, T. P., & Victer Paul, P. (2019). Optimal stochastic gradient descent with multilayer perceptron based student's academic performance prediction model. Recent Advances in Computer Science and Communications. https://doi. org/10.2174/2666255813666191116150319.

[9] Liu, Sainan & Pan, Haoan. Rare itemsets mining algorithm based on RP-Tree and spark framework. AIP Conference Proceedings. 1967. 040070. 10.1063/1.5039144, 2018.

[10] Krishan Kumar Sethi, Dharavath Ramesh , " HFIM: a Spark-based hybrid frequent itemset mining algorithm for big data processing ", Springer Science+Business Media New York , Jan 2017

[11] Loganathan, J., Latchoumi, T. P., Janakiraman, S., & parthiban, L. (2016, August). A novel multi-criteria channel decision in co-operative cognitive radio network using E-TOPSIS. In Proceedings of the International Conference on Informatics and Analytics (pp. 1-6).

[12] Loganathan, J., Janakiraman, S., & Latchoumi, T. P. A Novel Architecture for Next Generation Cellular Network Using Opportunistic Spectrum Access Scheme. Journal of Advanced Research in Dynamical and Control Systems,(12), 1388-1400.

[13] Zhang Feng, Liu Min, Gui Feng, Shen Weiming, Shami Abdallah, Ma Yunlong, "A distributed frequent itemset mining algorithm using Spark for Big Data analytics", Cluster Computing,18, 1493-1501, 2015.