RESEARCH ARTICLE OPEN ACCESS

A Survey on Optimal Job Scheduling Algorithms for Data mining in Cloud Environment

[*1]D.S.Misbha, [2]Dr. J. R. Jeba
Assistant Professor, Department of Computer Applications, Nesamony Memorial Christian College, Tamil Nadu, India.
Associate Professor and Head, Department of Computer Applications, Noorul Islam Center for Higher Education, Kanyakumari, Tamil Nadu, India.
Corresponding author: D.S.Misbha

ABSTRACT
Scheduling refers to a set of mechanisms and policies to control for the order of work that to be performed by a computer system. Scheduling is the method in which processes or threads, data flows are given right to use to system resources (e.g. communications bandwidth, processor time). This is frequently done to load stability to a system to perform successfully or achieve an intention to the excellence of services. The necessitate for a scheduling algorithm arises from the conditions for most contemporary systems to execute multitasking (can implement more than one process at a time) and multiplexing (broadcast numerous flows concurrently). This paper surveys on different job scheduling algorithms.

Date of Submission: 11 -05-2017 Date of acceptance: 19-08-2017

I. INTRODUCTION

Scheduling restraints are algorithms that can be used for allocating the resources among parties which are instantaneously and asynchronously demand for them. Scheduling restraints are used in routers (to handle packet traffic) in addition to operating systems (to contribute CPU time in the middle of both threads and processes), printers (print spooler), most embedded systems, disk drives (I/O scheduling), etc. The main intentions of scheduling algorithms are to reduce resource undernourishments and to make sure that the fairness along with the parties utilizing the resources. Scheduling compacts with the difficulty for deciding which of the marvelous requests is to be allocated resources. There are copious different scheduling algorithms for such processing methods. In this section, we introduce several of them.[1]

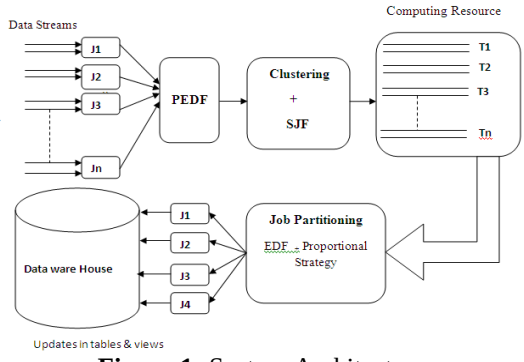

Figure 1: System Architecture.

In general, (job) scheduling is carrying out in three stages: short-, medium-, and long-term. In long-term (job) scheduling is made when a new process is created. It instigates the processes and so manages the measures of multi-programming (ie. amount of processes in memory). The long-term scheduler acknowledges more jobs when the resource consumption is low, and chunks the incoming jobs from entering the organized queue when utilization is too high.

Medium-term scheduling involves postponing or resuming methods by swapping (rolling) them out of or into the memory. When the central memory becomes over-assigned, the medium-term scheduler discharges the memory of a suspended (blocked or stopped) process by swapping (rolling) it out. Short-term scheduling arises most recurrently and decides which method to implement next. Short-term scheduler, also recognized as the process or CPU scheduler that controls the CPU sharing between the ‘‘ready’’ processes. The choice of a process that is to be executed next is done by the short-term scheduler. The main aspiration of short-term scheduling is to optimize the performance of the system, and however make available approachable service. [2]

II. SCHEDULING ALGORITHMS

2.1. Global Fair Lateness (G-FL) Scheduling Algorithm

G-FL is a G-EDF-like scheduler, but has lesser maximum tardiness bounds than GEDF. G-FL

is a GEL like scheduler that offers the same lateness bound (under CVA) for all jobs. For each job, G-FL uses a PP that go before its closing date. G-FL at rest makes available a reasonable distribution of lateness bounds to all jobs. G-FL is considerably better than G-EDF both in provisions of systematic bounds and in terms of practical delay. G-FL provides equivalent tardiness bounds for all jobs in the system. G-FL preserves the desirable JLSP (Job Level Static Priority) possessions.[3]

2.2. Partitioned-Distributed-Deadline Monotonic Scheduling Algorithm

Partitioned/Distributed-Deadline Monotonic Scheduling (P/DDMS) algorithm formulates the utilizes of the DST model for scheduling parallel/distributed fixed-priority fork-join real-time everyday jobs. The P/D-DMS algorithm is revealed to have a resource amplification bound of 4, which means that any job set that is realistic on m unit-speed processors and a single shared bus valid-time network, can be scheduled by any of this algorithm on m processors and a single communal bus real-time complex environment that are 4 times faster. The P/D-DMS algorithm is the transmitting algorithm for separating the set of tasks on top of the elements of the dispersed organization. [4]

2.3. Particle Swarm Optimization Algorithm

It is an additional method that optimizes a difficulty by iteratively attempting to develop the potential solutions with stare to a specified measure of eminence. PSO optimizes a problem by having inhabitants of possible solutions, here dubbed elements, and moving these specks around in the state-space according to the particle's location and rapidness. [5]

Each particle's movement is manipulated by its limited best known location and is also directed toward the best recognized positions in the search-space, which are then modernized as improved positions are found by other particles. This is expected to move the cloud toward the best clarifications. PSO algorithm and its constraints must be chosen so as to suitably balance between examination and utilization to avoid precipitate convergence to a local optimum nevertheless still ensure a good rate of convergence to the optimum. [5]

2.4. Multi-track Scheduling Algorithm

When the updates for the table appear on the stream, it is adapted into jobs. These jobs are then arranged according to their arrival time using PEDF algorithm. The sorted list of jobs is then partitioned into appropriate gathers depending on some criteria. (Such as jobs having the arrival time between 1 to 5 goes on to the first cluster and so on). All the jobs within one groups are sorted according to SJF (Shortest job first) algorithm of their implementation time and for the typical data. [6]

The computing resource is then reasonably partitioned into numerous tacks. Traditional way to make certain resource allocation is job partitioning and then scheduling the partitions independently. The up to date result indicates that by overall scheduling shows better performance accomplished in real time environment. By combining a quantity of features of these two strategies the multi-track algorithm verifies that the performance is better than using only EDF partitioning Strategy. When all the jobs are scheduled & implemented by multi-track algorithm the data warehouse updates are done instantaneously.[6]

2.5. Multi-track Proportional Algorithm

The Multi track Proportional algorithm is for scheduling the huge and assorted job sets come across by a streaming warehouse. This algorithm was proposed to handle the multifaceted atmosphere of a streaming data warehouse. Then a scheduling framework was planned that allocates jobs to the processing tracks and uses essential algorithms to schedule jobs inside a track. The main characteristic of framework is the capability to set aside resources for short jobs that frequently corresponds to important commonly refreshed tables, while avoiding the inefficiencies correlated with partitioned scheduling techniques. [7]

2.6. Max Benefit basic Algorithm

This algorithm is proposed to reduce data staleness ultimately. It is to handle the difficulties encountered by a stream warehouse: view hierarchies and priorities, inability to pre-empt, updates data reliability, heterogeneity of update jobs origins by different inter appearance times and data capacities among different sources and transitory overload. Max Benefit basic algorithm is comparable to the max-impact algorithm.[8]

2.7. Max Benefit Update Scheduling Algorithm

This algorithm evokes the goal of the scheduler which is to minimize the heaviness staleness. In this circumstance, the most benefit of executing a given job may be defined by its priority weighted freshness delta (decrease in staleness). Similarly, the marginal benefit of executing the job is its benefit per unit of execution time. A natural online greedy heuristic is to order the jobs by the marginal benefit of executing them. This heuristic is referred to as Max Benefit. Since marginal benefit does not depend on the period Max Benefit is used for periodic and a periodic update jobs. One may argue that Max Benefit ignores useful information about the release times of future jobs. This algorithm is used to minimize the weighted staleness.[9]

2.8. Service Request Scheduling Algorithm and Optimum Scheduling Algorithm

The scheduling progression can be examined as a service request scheduling and the resource scheduling such as service provider and the resource provider. The users can put forward their consequent request for implementing their appliances that contains one or more services to the Service Provider. At the moment the Service Provider has to carry out the service request scheduling process with these requests and has to activate on a massive set of data. Consequently the Service Provider necessitates a scheduler to resourcefully schedule these request and capitalize on the QoS to the consumer and the profit on the Service Provider site. Hence the method of service scheduling establishes here. Every request will be fixing together into task units and are dispensed with some accidental priority into the classifier. The classifier move forward these job components into a suitable scheduler units based on the situation of the scheduler units. The scheduler units accomplish the job component rooted in a few algorithms. These algorithms consider the consumption ratios as the deciding issues for priority relocation. [10]

The optimum scheduling algorithm works as follows. The arriving jobs to the agents are clustered on the basis of their category- deadline constrained or low down price constraint. After the preliminary clustering they are prioritized according to the target or income. This is necessary since the jobs with shorter deadline need to be scheduled first and similarly the tasks resulting in more profit ought to be scheduled on misplaced cost machines. Thus, the prioritizing constraint is special based on the environment or the kind of job. [10]

2.9. Cuckoo Search Algorithm

The Cuckoo Search Algorithm was established to reduce the overall completing time of corresponding jobs in a cloud system. The first step of the Cuckoo Search Algorithm is to defining inhabitants and the data demonstration. The populations are initialized by a suitable vector, wherein the extent of the vector point outs the quantity of resources. [11] The Cuckoo Search Algorithm is a novel meta-heuristic approach that represents the ordinary performance of the cuckoos. Every consumer brings up to dates single job at an instance and stores it in an arbitrarily preferred cloud. The most excellent cloud among high quality jobs are approved for the execution process. The total numbers of accessible host in the clouds are fixed. The number of host cloud can recognize by an unauthorized job by a possibility. The host customer can be either eliminate the job or depart the cloud with the intention to create a fresh cloud in a fresh location.

2.10. Energy-Aware Rolling-Horizon Scheduling Algorithm

The energy-efficient scheduling algorithm is a heuristic approach. This algorithm assigns the entire jobs to a Virtual Machine in a way to destructively assemble the jobs to its targets while keeping the power utilization. The energy-efficient scheduling algorithm estimates the jobs with its beginning instant and the implementation time on each Virtual Machine. If the jobs' target can be fulfilled while symbolizing these jobs can be assigned, after that the algorithm computes the jobs' power utilization. If the job may not be successfully assigned to any existing Virtual Machine, it calls the Function ScaleUpResource() motivating to provide accommodation to the job by increasing supply. If the jobs can be assigned, at that moment it chooses the Virtual Machine by giving up least power utilization to carry out the jobs, or else, the algorithm discards the jobs. When the jobs may not be successfully assigned to any existing Virtual Machine, the function ScaleUpResource() is called to produce a new Virtual Machine with the purpose of completing the job within its time limit.[12]

2.11. Inter-Cloud Meta-Scheduling (ICMS) Algorithm

The Inter-Cloud Meta-Scheduling (ICMS) Algorithm is rooted in a new message swap method to tolerate the optimization of job scheduling process. The ensuing system offers the advanced elasticity, sturdiness and delegation. This Inter-Cloud Meta-Scheduling algorithm is the means to symbolize the inter-cloud service allocation that permits the assimilation of modular guidelines. The ICMS is organized in a layered arrangement. The most important functionalities are separated in three layers explicitly, the service submission layer, the distributed resource layer and the narrow reserve organization layer. In the layer 1, a predefined topology that comprises the users that promotes the request to layer 2. The second contains a arbitrary topology based on the random interconnections of disseminated meta-brokers which represented as a nodes to swap over services. The service distribution is derived from messages that are exchanged in the middle of the meta-brokers. The ICMS holds a energetic workload administration to tolerate decision-making for services circulation on the meta-brokering altitude.[13]

2.12. Application-Level Scheduling Algorithm

An inferior bound was offered on the target violation prospect for application-level scheduling with the deadline limitations below a specified network capability. This inferior bound is tensed in the large-system establishment as it can be accomplished by suitably intended scheduling methods. This consequence clasps under extremely

all-purpose control models that may possibly have several transmission charge and still sequential association patterns. A novel scheduling strategy, called Maximum-Total-On-users (MTO) was introduced. This strategy was not just asymptotically most favorable in the large-system management, but furthermore accomplishes better performance for average-sized systems. It simplifies the consequences from single-class systems to multi-class systems, wherever the performance constraints of dissimilar classes can be at variance considerably. Further, founded on the exceeding asymptotic advance, the Application-Level Effective Capacity (ALEC) has been considered, i.e., the utmost throughput that can be maintained by the system with the given constraints on the deadline contravention probability. [14]

2.13. Priority Based Job Scheduling Algorithm

This new scheduling algorithm is based on multi-criteria and multi-decision priority. In this algorithm the given task is divided into three levels: alternate level, object level and attribute level. The priority of the task can be set in this algorithm based on job resource ratio. With the priority vector the task is compared with the queue. The result of algorithm shows the higher throughput and less finish time.[15]

2.14. An efficient Multi Queue Job Scheduling

In MQS (Multi Queue Scheduling) scheduling algorithm, first of all, the tasks are assigned in ascending order and then it is divided into the relevant of medium, small and large size queue. Then the Meta scheduler allocates the entire given task to the virtual machine. The result of this algorithm illustrates the increases user satisfaction and utilizes the free unused space so that the performance is increased.

2.15. Fair4s Job Scheduling

Fair4S is intended to be influenced towards small jobs. Small jobs are accounted for the best part of the workload, and the majority of them necessitate immediate and interactive responses, which is a vital phenomenon at the production of Hadoop systems. The inefficiency of Hadoop fair scheduler and GFS read write algorithm for managing small jobs inspires us to employ and analyze Fair4S, which initiates pool weights and extends job priorities to assurance the rapid responses in favor of small jobs [17]. The execution of Fair4s Job Scheduling Algorithm runs on a huge cluster of service machines and is elevated scalable. Map-Reduce is admired by open-source Hadoop project. The Fair4s Job Scheduling Algorithm works on the dispensation of large records by dividing them on the amount of chunks and allocating the tasks to cluster nodes in Hadoop multimode configuration. In these ways the Fair4s Job Scheduling Algorithm improves the consumption of the Cluster nodes in terms of storage, CPU, and Time.

2.16. Job Scheduling and Optimized Genetic Algorithm

Genetic Algorithm (GA) is derived from the biological concept of generation of the population, a fast growing area of Artificial intelligence Genetic Algorithm's are motivated by Darwin's theory regarding evolution. According to Darwin "Survival of the fittest", is used as the technique of scheduling in which the jobs are allocated to respective resources according to schedules in circumstance of scheduling, which informs with reference to which resource is to be assigned to which job. Genetic Algorithm is based on the biological idea of population generation.

Scheduling by make use of Genetic Algorithm (GA) is resultant from the biological theory of generation of the population, a rapid increasing area of Artificial intelligence, Genetic Algorithm's are motivated by Darwin's theory regarding evolution. Genetic Algorithm is based on the biological idea of population generation which sequentially proposed as a resolution for Multi-objective optimization for effective resources. When a demand is made for any resources then the efficient resource scheduling is mapped on top of substantial resources with appropriate load balancing which in turn very difficult to accomplish. This algorithm is in estimation with random, grade and static algorithms. The process of virtualization take place between the users and the physical layer and it has three characteristics usability, secure and moving. The virtual resources are abstracted by making number of incidences of actual physical resource nodes with elements. This algorithm is regarded to be heuristic so that it contains entity utilities, code and exploring methods. Object functions are required for load balancing calculation. The GA has assortments, intersect and mutation. In this algorithm the selection is tournament selection, crossover is two point crossovers and in the mutation is if the random number is being chosen that is the unique gene is restored by randomly generated one. By this taking into consideration the CPU handling, memory and bandwidth the NDSA II comes out to be a better algorithm than a rank, random and static algorithm as it provides numerous selections by running just once resourcefully.[18]

2.1.7 SMine Algorithm

It is to find the frequent itemsets. The database is sorted based on the number of transactions. Thus it is named as SMine algorithm. The number of scans and time are reduced to find the number of itemsets in the database. It accepts only nominal attributes and it is well for finding frequent itemsets.

2.1.8 Modified SMine Frequent Itemset Mining Algorithm

This algorithm is also used to find the frequent itemsets. It is the extension of SMine Algorithm. In the modified SMine, the database is not sorted based on the number of transaction. Its performance is better than SMine algorithm.

III. CONCLUSION

Job scheduling algorithms always play a crucial role in data mining. Scheduling is a major problem in data mining. In this paper, various job scheduling algorithms related to data mining were surveyed and studied. The main aim of job scheduling algorithms is to maximize the resource utilization and to convince the user requirements. Scheduling algorithms are implemented in order to improve the user submission along with the virtual machines. Starvation problem is created when priority is considered in job scheduling. Hence, there are many features of study based on priority and improving parameters like deadline, execution time, overall performance and average resource utilization ratio. This survey has provided a clear idea about various job scheduling algorithms and their functions.

REFERENCES

[1]. Masoud Nosrati, Ronak Karimi and Mehdi Hariri, "Task Scheduling Algorithms Introduction," World Applied Programming, vol. 2, issue. 6, pp. 394-398, June 2012.

[2]. Eskicioglu and Marsland, "Scheduling," Jan 2001.

[3]. Jeremy P. Erickson and James H. Anderson, "Fair Lateness Scheduling: Reducing Maximum Lateness in G-EDF-like Scheduling," Real-Time Syst, vol. 50, issue. 1, pp. 5-47, July 2013.

[4]. Ricardo Garibay-Martínez, Geoffrey Nelissen, Luis Lino Ferreira and Luís Miguel Pinho, "On the Scheduling of Fork-Join Parallel/Distributed Real-Time Tasks," Ninth IEEE International Symposium on Industrial Embedded System, pp. 31-40, June 2014.

[5]. Boopathy S.S. and Subramanian K.M., "Improved Scheduling and Minimized Updates in Data Warehouses," International Journal of Advanced Research in Data Mining and Cloud Computing, vol. 1, issue. 2, August 2013.

[6]. Rane V.N. and Makhijani R.K., "Updating Streaming Data Warehouse by Scalable Scheduling using Multitrack Algortithm," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 2, issue. 6, June 2014.

[7]. Subhani S.M and Srinivas Reddy G., "Dynamic Updates on Streaming of Data ware Houses Explore Tradeoffs," International Journal for Development of Computer Science and Technology, vol. 1, issue. V, Aug-Sep 2013.

[8]. Sreeja A., Sailaxmiharitha I.V. and Bhaskar N., "Load Balancer Scheduling Over Streaming Data in Federated Databases," International Journal of Engineering Research and Technology, vol. 2, issue. 8, August 2013.

[9]. Bolla Saikiran and Kolla Morarjee, "An Efficient Algorithm for Update Scheduling in Streaming Data Warehouses," International Journal of Computer Science and Information Technologies, vol. 5, 2014.

[10]. Ram Kumar Sharma and Nagesh Sharma, "A Dynamic Optimization Algorithm for Task Scheduling in Cloud Computing with Resource Utilization," International Journal of Scientific Engineering and Technology, vol. 2, issue. 10, pp. 1062-1068, Oct. 2013.

[11]. Nima Jafari Navimipour and Farnaz Sharifi Milani, "Task Scheduling in the Cloud Computing Based on the Cuckoo Search Algorithm," International Journal of Modeling and Optimization, Vol. 5, No. 1, February 2015.

[12]. Xiaomin Zhu, Laurence T. Yang, Huangke Chen, Ji Wang, Shu Yin and Xiaocheng Liu, "Real-Time Tasks Oriented Energy-Aware Scheduling in Virtualized Clouds," IEEE Transactions on Cloud Computing, vol. 2, no. 2, April-June 2014.

[13]. Stelios Sotiriadis, Nik Bessis, Ashiq Anjum, Rajkumar Buyya, "An Inter-Cloud Meta-Scheduling (ICMS) Simulation Framework: Architecture and Evaluation," IEEE Transactions on Software Engineering, 2015.

[14]. Huasen Wu, Xiaojun Lin, Xin Liu and Youguang Zhang, "Application-Level Scheduling With Probabilistic Deadline Constraints," IEEE/ACM Transactions on Networking, 2015.

[15]. Shamsollah Ghanbari,Mohamed Othman," A Priority based Job Scheduling Algorithm in Cloud Computing" ICASCE, 2012.

[16]. Karthick A. V., Ramraj E., Ganapathy Subramanian R., "An Efficient Multi Queue Job Scheduling For Cloud Computing", IEEE, 2014.

[17]. ZujieRen, Jian Wan "Workload Analysis, Implications, and Optimization on a Production Hadoop Cluster: A Case Study on Taobao", Co IEEE Transactions On Services Computing, Vol. 7, No. 2, April-June 2014.

[18]. Deva Prasad K., Samatha V., Sambasiva Rao G., "Resource Allocation and Job Scheduling Using Genetic Algorithm in Cloud Computing Environment," International Journal of Advanced Research in Computer Science

Engineering and Information Technology, Volume: 4, Issue: 2, 02-Jan-2015.
[19]. [19]. Jeba J.R., Dr. S.P Victor "Comparison of Frequent item set Mining Algorithms" International Journal of Computer Science and Information Technologies, Vol.2(6), pp.2838-2841, 2011.
[20]. [20]. Jeba J.R., Dr. S.P Victor "Effective Measures in Association Rule Mining" International Journal of Scientific & Engineering Research, Vol 3, Issue 8, 2012.
[21]. Jeba J.R., "Comparison of Frequent item set Mining Algorithms," International Journal of Computer Science and Information Technologies, Vol. 2 (6) ,pp. 2838-2841

International Journal of Engineering Research and Applications (IJERA) is **UGC approved** Journal with Sl. No. 4525, Journal no. 47088. Indexed in Cross Ref, Index Copernicus (ICV 80.82), NASA, Ads, Researcher Id Thomson Reuters, DOAJ.

D.S.Misbha. "A Survey on Optimal Job Scheduling Algorithms for Data mining in Cloud Environment." International Journal of Engineering Research and Applications (IJERA), vol. 7, no. 8, 2017, pp. 60–65.