TIME SCHEDULING ALGORITHMS FOR LOAD BALANCING THROUGH OPTIMIZED HYBRID APPROACH: A REVIEW

[1]DEVANSHU DHAR SRIVASTAVA, [2]Rekha

[1]M. Tech. Scholar, [2]Assistant Professor
Computer Science Engineering
CBS Group of Institutions

***Abstract*: This paper explores the FCFS which utilised with a priority queue. Hybrid production techniques like FCFS and priority queues improve system performance. Despite knowing the system setup, we are building and testing three solutions to meet our specific requirements. FCFS and the priority concept do not work well together. To execute fast, we use a hybrid method. If the top priority task consumes more execution time than the other jobs in the priority queue, there is no way to speed up the system. As this paper introducing a new system lowers the performance of all current systems. However, it is allocated automatically based on current system characteristics. Priority queues may enhance system performance in the future.**

***Keywords*: Scheduling Algorithms, Load Balancing, Optimized Hybrid Approach**

I. INTRODUCTION

The algorithms based on the FCFS to tasks the which they will be received. When it comes to making efficient use of the K-resource system's separately located resources, a job allocation method must be able to select any work based on the best match between job and that are accessible. To map six tasks' bytes work assignment method must take into consideration the following factors: This define the amount of CPU and RAM that each job will need. work reflects in which tasks are completed. As result of these assumptions, an assignment scheme would choose a number of tasks to be completed during the planning time. When comparing different ways of allocating work, the number of times it takes of the jobs in the queue is taken into consideration. The tasks for each planning period, uncontrolled workload, as well as the tasks for each planning period for FCFS choose any job from task that is appropriate era. Despite the fact that well-conceived demonstrates limitations job distribution systems on FCFS as well as the potential of less limited employment allocation methods in the future. The FCFS's allocation technique distributes jobs 0 and 1 in the first schedule period, but it is unable to assign employment 2 in the next schedule period since the total CPU demand of the three jobs exceeds the amount of CPU capacity available on the system. During the first period, FCFS/FF is circumventing this reward by bypassing jobs 2 and planning tasks 4 and 5, respectively. Tasks 2 and 3 must, on the other hand, be scheduled in separate epochs due to the fact that no other jobs are available during each of these epochs. After everything was said and done, the approach clever avoid scheduling at time. are discovered that are perfectly matched to the machine configuration. As a result, the unconstrained system for the distribution of tasks periods of tasks.

Systems on FCFS/FF technology backfill machines in a greedy manner by selecting the next job, which is subject to backfill limitations. Most of the time, these methods do not take into consideration task pool tasks or the current system resource load state. In the K-resource planning problem, this approach results in the early depletion of some resources, while others stay underused as a result of the strategy. tasks and causes to be depleted much more quickly than the available RAM resources. In this paper, we aim to provide K-resource knowledgeable job selection heuristics that take into account additional needs for work resources as well as system circumstances to affect the selection of the next task to be performed. They can be easily included into the FCFS/FF scheduling algorithms that are already in use. We discovered that the improved FCFS/FF method for the scheduling of K-resource-conscious heuristics outperforms the prior greedy, first-fit heuristics in terms of average system response time when compared to the prior greedy, first-fit heuristics.

Job scheduling is used to assign certain tasks to specific resources. In a network system, task scheduling is a major and difficult topic. The job planner should thus be dynamic. Job scheduling in network systems is primarily intended to enhance the effective use of bandwidth, memory and time-reduction resources. An effective work scheduling strategy must be designed to provide less time to respond in order to ensure that the work supplied is executed within a feasible minimal time period and that resources are re-allocated in-time. This reduces job rejection and allows customers to submit more tasks to the cloud, which eventually shows an improvement in the business performance of the cloud. Various scheduling modes are described below according to different requirements, for example static vs dynamic, centralized versus distributed, offline versus online, etc.

1. Static Timing: Pre-schedule jobs, all available resources and work information is known and a task is once allocated to a resource, thus it is simpler to adjust from a scheduler point of view.

2. Dynamic Scheduling: The scheduler may dynamically plan jobs over time. It is flexible to determine run-time in advance than static scheduling. The key element for obtaining reliable, accurate and efficient scheduler algorithms is the inclusion of load balancing.
3. Centralized programming: As stated in dynamic planning, the centralized/distributed planner is responsible for making global decisions. The main advantages of centralized scheduling are easy installation, efficiency and greater resource management and monitoring. On the other side, this programmer lacks scalability, tolerance to failure and efficient performance. It is not recommended for big grids because of this drawback.
4. Distributed/Decentralized Scheduling: this form of scheduling is, despite its low efficiency, more realistic than centralized scheduling for the actual cloud. There is no central control body, thus requests by local schedulers to manage and maintain the employment queue
5. Pre-Emptive Scheduling: this kind of scheduling enables every task to be stopped during execution and permits the migration from the initially allotted resource to a different resource for other jobs. This kind of scheduling is more useful if restrictions such as priorities are addressed.
6. Non pre-emptive scheduling: it is a scheduling method in which resources cannot be reassigned before the work runs and its planned tasks are completed.
7. Co-operative scheduling: the system already has a large number of schedulers in this kind of scheduling and is each accountable for carrying out specific activity in planning towards a large range of common system based on cooperation processes, regulations and existing system users [9].
8. Immediate/Online mode: scheduling schedules every newly arrived task as soon as it comes in without waiting for the next time on available resources. 8.
9. Batch Mode: the planner saves the coming tasks as a collection of issues which have to be resolved across time intervals, so that the work may better be mapped according to its features for enough resources.

II. LITERATURE REVIEW

There are many research articles on load balance that demonstrate different load balancing techniques for a basic network or distributed network. The most significant problem is load balancing. Some research articles are provided below to investigate load balancing.

In this article Masoud Nosrati Ronak Karimi Mehdi Hariri [1] suggested that the scheduling of operating systems be implemented. Main approaches and scheduling strategies are given and briefly explained. Thus, the key ideas are shown with a brief introduction describing the long-term, medium-term, short-term and dispatching schedule. Investigated techniques include: FIFO, shortest first job, pre-emptive priority planning, round-robin and multi-level feedback queue. These approaches are finally contrasted.
According to this research[2] Grid computing expands with a computing platform that collects linked a spread to create Calculation addresses difficult computation issues many computers. resolves requirements environments. primary focus devoted of resources task planner. aim work optimize use of resources reduce working grid scheduling methods place little focus on the performance of a grid planner in the processing time parameter. Scheduler assign resources to the work that is to be performed utilizing the First Coming Algorithm. In this work, they developed an optimized algorithm for the scheduler to queue utilizing several scheduling techniques such as Shortest Job First, first in first, round robin. The job scheduling system is responsible for choosing the best machines for user work. programming produces work every into account constraints characteristics tasks. primary objective article create effective work planning optimize use to reduce work. Ques may employing different according to , performance. was performed ASP.NET.
Researchers are explaining a novel method to planning CPU algorithms that may be utilized to enhance the ETS[3]. The novel method depends integration the determined of quantity maintains basic roundrobin to reduce hunger and also incorporates the value of priority planning. The suggested method additionally incorporates the SJF and time quantum idea by setting the processes new priority. Existing round robin CPU programming method cannot be applied in real time because of their huge waiting time, extensive reaction time, and larger turnaround times and worse performance. With the new approach method, the round robinCPU scheduling technique overcomes all disadvantages. It also provides the comparative analysis of the suggested method of the the time, and the.

The issue of real time planning covers a wide range of methods from basic uniprocessor to very advanced multiprocessor scheduling algorithms[4] is discussed by a research team. They examine the features and limitations of real-time activities to be performed in this job. Analysis techniques and the notion of optimality criteria are also discussed, which lead to the creation of suitable planning algorithms. They then examine algorithms for uniprocessor systems that may be split into twomajor classes: off-line and on-line. Online algorithms are divided into static algorithms based on dynamic priority. They describe algorithms based both on preventive and non-preemptive priorities. They investigate thetwo subgroups in dynamic priority-based algorithms; namely, planning and best effort planning algorithms. Some of the programming algorithms of the uniprocessor are shown in the Appendix. Another type of real time scheduling methods, also addressed in the article, are multiprocess scheduling algorithms. They also discuss methods to address aperiodic and irregular jobs, limitations on precedence and priority reversal.

A novel CPU method named A New CPU Scheduling Algorithm[5] is presented in this article, which operates as both preemptive and non-preventive on the basis of arrival times. The supposed method helps increase the CPU efficiency of the operating system in real-time. The foundation of the multi-programmed operating system is CPU scheduling. The programmer is responsible for CPU multiplexing tasks. A multi-programmed operating system has various scheduling algorithms available as FCFS, SJF, Priority,

Round Robin, etc. The results of the current algorithms (FCFS, SJF, Priority and Round Robin) are compared with the method presented in this article.

Computer systems serve various functions; without them, they may envision a world. It thus enables us to perform things at tremendous rates and it offers up a new universe of possibilities to explore[6]. One of the most essential computer systems aspects is the capacity to perform many tasks at once, or at least the perception of the excellent scheduler. In order to accomplish this, the system switches between processes rapidly. Each process is performed for a given period and when the schedule is out of the process it selects to run next on the CPU for a particular length of time. The behavior of a system thus relies mostly on the scheduler. The scheduler has to combine fairness and efficiency with the right behavior. In this article, they have suggested a novel version of an MLFQ method, in which time is allocated to each MLFQ schedule so as to modify the time slice dynamically and to optimize the turn-off time with each execution cycle. It should be noted that the overall performance of the Algorithm is enhanced by utilizing the dynamic quantum and neural network MLFQ using the static time slice of every queue.
A group of several researchers[7] investigate builders of real-time systems frequently without considering alternatives via priority planning in their systems. This article investigates a pre-run time scheduling alternative and shows that the application offers substantial benefits over priority scheduling methods.

Some scientists[8] have characterized the main aim of this work as the development of an improved method for current algorithms for priority planning. The method suggested significantly helps to reduce, decreasing procedures. The operates range operations, including, resources, activities, the allocation of resources is very important.

Many scheduling methods are available they concentrate attributes such as, , Medium Hunger. suggested method significantly lowers the current Priority Scheduling architecture as regards the degree to which lower priority processes are starved. The method adapts current performance characteristics to an optimum level, resulting in efficient outcomes for certain situations. In this article, they also conducted a comparison analysis of the current technology and the method suggested to show substantial performance parameter differences.

In 2005, a research[9] gave information on preventive ability as a required prerequisite to create real-time software. In comparison with non-preventive planning, there are extra expenses associated with preventive scheduling. In addition, the feasibility of a task set with non-preemptive programming does not indicate viability with preemptive scheduling in the context of the fixed priority planning (and vice-versa). In this article, the preventive threshold is used to create a novel scheduling model that integrates the ideas of preventive and non-preventive scheduling, both as specific instances.
Express Logic, Inc. developed the concept of prevention threshold in their Thread X real-time operating system. It enables a task to deactivate task prevention up to a defined priority level. Task priority over the threshold may still be prevented. Each job has a priority and a prevention threshold in our new approach. They demonstrate how reaction times may be evaluated under the new scheduling paradigm. They also create algorithms for optimum prioritization and prevention.
In this article they also offer proof of the significant quantitative advantage of the novel scheduling model over preemptive and non-preemptive scheduling methods. We demonstrate that the new approach may lead to a significant increase in scheduling by using both preventive and non-preventive scheduling elements. In addition, the new model shows that it offers reduced run-time costs, avoiding needless preemptions.

Authors of this work[10] described in the age of multiprogramming system supercomputers have developed. The Multiprogramming operating system enables more than one process ready to be carried out in memory loaded. CPU planning is the processes that may be selected from among the memory processes ready to run and assign CPU time. For schedulingCPUs such as FCFS, shortest job first (SJF), priority scheduling etc, a number of common methods have been developed. But no algorithm is perfect in terms of more performance, less waiting time, lower turnaround time etc. This article proposes a novel fluffy logic-based PCU programming method to address limitations of traditional CPU efficiency strategies.
The performance of parallel programs such as those using fork joining instructions is greatly influenced by the technique used for programming jobs. This article [11] examines parallel task planning in uniform distributed systems. A simulation model is utilized to solve scheduling performance problems. Different rules are used to arrange parallel tasks for a variety of workloads. Fairness between competing jobs is needed. We investigate situations where the distribution of the number of parallel tasks per job and the distribution of the demand for task service vary with time. We also study the effect of overheads needed to acquire global system information on the performance of processor queues. Simulated findings show that, while all scheduling techniques are worthwhile, overall performance is substantially improved and fairness in terms of the individual performance is guaranteed.

Job planning is one of the main tasks in all computer systems. Cloud computing is one of the newest technologies that is dramatically developing. To improve the work of cloud computing environments effectively, work planning is one of the responsibilities to maximize profit. In this article we present a multi-demand scheduling system. A web application is created that offers two kinds of service downloads for authorized users. The use of non-preventive prioritization algorithms handle multiple requests. The primary objective of the service provider is to offer rapid services for numerous requests. This article provides the appropriate approach and technique to get optimistic service value. This study examines the quality of service objectives of consumers and service providers[12]. The system's primary goal is to obtain an affirmative reaction at the conclusion of users. The use of resources is done very temporarily. The experimental investigations indicate that there are promising outcomes in the suggested system.

In the 1990s, the word "Grid" was created to indicate a dispersed, sophisticated scientific and engineering infrastructure. Grid is a mechanism that distributes computers among different organisations. Grid computing is a distributed type of computation. It includes pooling resources that are diverse and dispersed globally to solve different difficult issues and create large-scale applications. Resources are needed for the development of applications and thus job and resource planning is a major topic of study in grid computing. The aim of the programming is to achieve maximum system performance and to meet the demands for the application with the computing resources available[13]. In this article, they will examine the different work and resource algorithms that assist academics in carrying out future research in this field.

When assessing the performance of a complex system, numerous decisions have to be made. In the simultaneous planning of the task, it is necessary to determine what workload to utilize and which measures to take. Sometimes these choices have subtle consequences, which may be easily overlooked. With this article, several hazards may be found, in the aim of at least helping to prevent them[14]. They also suggest areas that might benefit from further study.

For complexity, unpredictability and dynamic features, the issue of work plan is recognized as NP-hard, and many investigations have been performed in the past on scheduling algorithms. This article first identifies and classifies the issue of work schedules and then performs a systematic and comprehensive comparative analysis of work scheduling algorithms on the basis of their origin, characteristics, calculation and use[15]. Finally, certain trends and features of work planning algorithms are suggested.

Task planning algorithm study is one of the main grid computing methods. First of all, this article dewrites a DAG task planning model for grid computing, secondly covers Generational Planing (GS) and Communications Inclusion Generational Planning (CIGS) methods [16]. Finally, an improved CIGS algorithm is suggested for usage in grid computing and has been successfully demonstrated.

This article offers a load balancing method modelled for swarm intelligence [17] for grid job planning. ParticleX will discover optimum or almost optimal solutions in issue search space. known as distributes working burden amongst node. The the algorithm. A number of iterations are performed to find potential optimum solution for the planning of resource work. At every iterative phase a solution is created and the maintained work, utilized a for -planning Experimental investigations indicate that the new method suggested is - schedules.

Cluster computing is a cost efficient parallel processing method to solve various difficult computer problems. Middleware solutions that handle rules, protocols, networks and work schedules among linked computing resources are the key to making the cluster computing function properly. The research issue addressed in this article is the difficulty with multi-cluster scheduling online. In order to do this, we offer an online dynamic scheduling strategy that administers many workflows across single and multiple cluster computing systems with the aim to improve the average response time and system utilization[18]. The implementation of the suggested scheduling strategy is contrasted with a policy on space sharing and time sharing. The results of the tests indicate that the strategy suggested results in substantially better times than the two other strategies.

The Flexible Job-shop Scheduling Problem (FJSP) provides the time and technical requirements for the execution of tasks by a number of candidate resources. This project is based on an analgorithm that resolves every) using aparticular functions performing a set to schedulation [19]. This work follows hierarchical architecture The goal paper examine effect several characteristics restriction reach greatest possible level of objectives. To show the effectiveness of this method, experiments have been carried out and the GA parameters properly tuned to comparable situations.

MinimizingWaiting Time Variation (WTV) is a work schedule issue, in which a batch of n tasks are scheduled to be serviced by the single source, such that their time variance is minimised. Minimizing WTV is an essential planning issue, in many sectors, in delivering quality of service (QoS). Minimizing task waiting time variations on computer networks may result in consistent and predictable network performance. Since NP-hard is the WTV minimization issue, they propose two heuristic work schedulation techniques named Balanced Spiral and Verified Spiral with proved features of optimum job sequences for this problem. We evaluate and compare our techniques with four existing ways of scheduling minor and big issue cases. The findings of the performance indicate that Verified Spiral provides the greatest performance for the techniques and issues evaluated for this study[20]. Balanced spiral outcomes are similar, but at lower costs. During our examination, we found a consistent pattern in the WTV plot for all conceivable sequences of a series of tasks that may be utilized to assess the sacrifice of mean waiting time during the pursuit of WTV minimisation.

Multiprocessor task planning is a critical and computer-driven problem in modern computing. The genetic algorithm and algorithm scheduling are both discussed in this article. Also techniques are fundamentally parallel, yet they are both reliant on a large amount of data. On the basis of experimental results, this article offers an in-depth, downsides technique discussed in detail. Multiprocessors have been created as a powerful computer tool for real-time applications, particularly in cases when a single processor system is insufficient to handle all of the required tasks. Because of the high performance and long life of multiprocessors, they have become a significant computational resource. When working in such a computer environment, an efficient method is required be completed. The ability to efficiently arrange a parallel programme on the processors in multiprocessor systems in order to minimise the overall runtime is critical for achieving high performance. This problem has been classified as NP-Hard. When it comes to multiprocessor scheduling, a programme should for a certain in a way minimised. quickly is reasonably. When it comes to limited optimization, the most often (GA). Biological and [21], in its most basic form, are search algorithms based on natural and genetic mechanics. List

scheduling techniques prioritise each item that needs and organise descending order of importance. When enough processors are available, the job with the greatest priority on the task list will be processed and removed from the list of open jobs. When been assigned applicants are chosen at random from the pool of candidates. It is discussed in this article how genetic algorithms (GA) and lists heuristic scheduling (LSH) may be used to solve multiprocessor scheduling problems.
This article offers a Priority (SAP)-based algorithm based on a school's learning resources and video-on-demand services in schools, creating the cloud environment, and analyzing the video data storage of the cloud storage and administration. The algorithm timetable on the supply and demand management of data blocks is comprehensive and precise, according to the data block priority, and resolves the system launch delay and the continuity of streaming media players.

III. CONCLUSION AND FUTURE SCOPE

FCFS and the priority notion are failing to provide excellent performance when executed independently. But a hybrid approach allows us to execute quickly and improve performance. In future, if any idea might be implemented on the basis of priority, then there is no function accessible to enable the system to work quickly if a top priority is a job but it consumes more execution time compared to other tasks at the priority queue. The biggest fault in this project is that it creates a small system such that adding a new system affects all systems performance. Priority must be defined at selection of tasks, but it automatically allocates priority without any explicit command on the basis of available system settings. In future, anybody may improve the performance via the use of the priority queue idea on systems according to their configuration.

REFERENCES

[1] Masoud Nosrati Ronak Karimi Mehdi Hariri, “Task Scheduling Algorithms Introduction” World Applied Programming, Vol. (2), Issue (6), June 2012
[2] Pinky Rosemarry, Ravinder Singh, Payal Singhal and Dilip Sisodia,“Grouping Based Job Scheduling Algorithm Using Priority Queue and Hybrid Algorithm In Grid Computing” International Journal of Grid Computing & Applications (IJGCA) Vol.3, No.4, December 2012
[3] Neeraj Kumar, Nirvikar,“Performance Improvement Using CPU Scheduling Algorithm-SRT”, International Journal of Emerging Trends & Technology in Computer Science, Volume 2, Issue 2, March – April 2013
[4] Arezou Mohammadi and Selim G. Akl,“Scheduling Algorithms for Real-Time Systems”, School of Computing, Queen’s University, Canada
[5] Sukumar Babu Bandarupalli1, Neelima Priyanka Nutulapati, Prof. Dr. P. Suresh Varma, “A Novel CPU Scheduling Algorithm-Preemptive & Non-Preemptive” International Journal of Modern Engineering Research, Vol.2, Issue.6, Nov-Dec. 2012
[6] Deepali Maste, Leena Ragha and Nilesh Marathe,“Intelligent Dynamic Time Quantum Allocation in MLFQ Scheduling” International Journal of Information and Computation Technology, Volume 3, Number 4 2013
[7] Jia Xu, David Lorge Parnas,"Priority Scheduling Versus Pre-Run-Time Scheduling" The International Journal of Time-Critical Computing Systems, 18, July 2000
[8] Siddharth Tyagi, Sudheer Choudhary, Akshant Poonia, “Enhanced Priority Scheduling Algorithm” International Journal of Emerging Technology and Advanced Engineering, Volume 2, Issue 10, October 2012
[9] Yun Wang and Manas Saksena, “Scheduling Fixed-Priority Tasks with Preemption Threshold" RTCSA’99, Hong Kong December 2005
[10] Prerna Ajmani and Manoj Sethi, “Proposed Fuzzy CPU Scheduling Algorithm (PFCS) for Real Time Operating Systems" International Journal of Information Technology, Vol. 5 No. 2, July-December, 2013
[11] Helen D. Karatza, “Parallel Job Scheduling in Homogeneous Distributed Systems”, Department of Informatics Aristotle University of Thessaloniki 54124 Thessaloniki, Greece
[12] Shalmali Ambike, Dipti Bhansali, Jaee Kshirsagar, Juhi Bansiwal, “An Optimistic Differentiated Job Scheduling System for Cloud Computing”, International Journal of Engineering Research and Applications (IJERA) Vol. 2, Issue 2, Mar-Apr 2012
[13] Pinky Rosemarry, Payal Singhal, Ravinder Singh, “A Study of Various Job & Resource Scheduling Algorithms in Grid Computing”, International Journal of Computer Science and Information Technologies, Vol. 3 (6), 2012
[14] Eitan Frachtenberg, Dror G. Feitelson, “Pitfalls in Parallel Job Scheduling Evaluation”, Modeling, Algorithms, and Informatics Group Los Alamos National Laboratory
[15] Yarong Chen, Zailin Guan, Xinyu Shao, “A comparative analysis of job scheduling algorithm” Management Science and Industrial Engineering (MSIE), 2011 International Conference on IEEE
[16] Liang Yu, Gang Zhou, Yifei Pu, “An Improved Task Scheduling Algorithm in Grid Computing Environment”, Int. J. Communications, Network and System Sciences, 2011
[17] Thanapal P, Dr. Gunasekaran G, “A Distributed Job Scheduling on the Grid Using Particle Swarm Optimization (PSO) Algorithm” International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 1, January 2013
[18] Abawajy and Dandamudi, “Parallel job scheduling on multi-cluster computing system” Cluster Computing, 2003. Proceedings 2003 IEEE International Conference
[19] Celia Gutiérrez, “Overlap Algorithms in Flexible Job-shop Scheduling” International Journal of Artificial Intelligence and Interactive Multimedia, Vol. 2, Nº 6
[20] NongYe, Xueping Li, Toni Farley, Xiaoyun Xu, “Job scheduling methods for reducing waiting time variance” Computers & Operations Research 34 (2007) 3069 – 3083

[21] Mrs S. R. Vijayalakshmi and Dr G. Padmavathi, "A Performance Study of GA and LSH in Multiprocessor Job Scheduling" IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 1, No. 1, January 2010
[22] Kunal Agrawal Yuxiong He Wen Jing Hsu Charles E. Leiserson, "Adaptive Scheduling with Parallelism Feedback" Singapore-MIT Alliance and NSF Grant ACI, March 29–31, 2006, New York
[23] Yeqing Liao, Quan Liu, "Research on Fine-grained Job scheduling in Grid Computing" I.J. Information Engineering and Electronic Business, 2009
[24] Xie Silian, "Research of Scheduling Algorithm Based on Priority in Data Nodes of Cloud Storage" Digital Manufacturing and Automation (ICDMA), 2011 Second International Conference of IEEE
[25] Ramamritham, "Scheduling algorithms and operating systems support for real-time systems" Proceedings of the IEEE (Volume:82 , Issue: 1), Jan, 2004
[26] Wang Xibo, "Research on Subsystem Hybrid Scheduling and Priority Inversion Based uCOS-II" Intelligent Networks and Intelligent Systems (ICINIS), 2012 Fifth International Conference of IEEE.