Efficient resource management techniques in cloud computing environment

Alka Kaushik

B. Tech, M. Tech MDU Rohtak
Email: cakapilraj@yahoo.com

Abstract- **Cloud resources and their loads possess dynamic characteristics. Current research methods have utilized certain physical indicators and fixed thresholds to evaluate cloud resources, which cannot meet the dynamic needs of cloud resources or accurately reflect their resource states. To address this challenge, this paper proposes a Selfadaptive threshold based Dynamically Weighted load evaluation Method (termed SDWM). It evaluates the load state of the resource through a dynamically weighted evaluation method. First, the work proposes some dynamic evaluation indicators in order to evaluate the resource state more. accurately. Second, SDWM divided the resource load into three states, including Overload, Normal and Idle using the self-adaptive threshold. It then migrated those overload resources to a balance load, and releases the idle resources whose idle times exceeded a threshold to save energy, which could effectively improve system utilization. Finally, SDWM leveraged an energy evaluation model to describe energy quantitatively using the migration amount of the resource request. The parameters of the energy model were obtained from a linear regression model according to the actual experimental environment. Experimental results showed that SDWM is superior to other methods in energy conservation, task response time, and resource utilization, and the improvements are 31.5 %, 50 %, 50.8 %, respectively. These results demonstrate the positive effect of the dynamic self-adaptive threshold. More specially, SDWM shows great adaptability when resources dynamically join or exit.**

Keywords— Cloud computing · Energy · Self-adaptive threshold · Dynamic weighted · Load evaluation

I. INTRODUCTION

In a cloud computing environment, many dynamic and uncertain factors relating to resources and loads can affect resource availability and task scheduling, e.g., a resource node load dynamically changes with time, and the amount of the resource request also changes with the year, the season and with holidays [1, 2]. Meanwhile, there are many changes in the resource itself, i.e., resources may join or exit at any time.

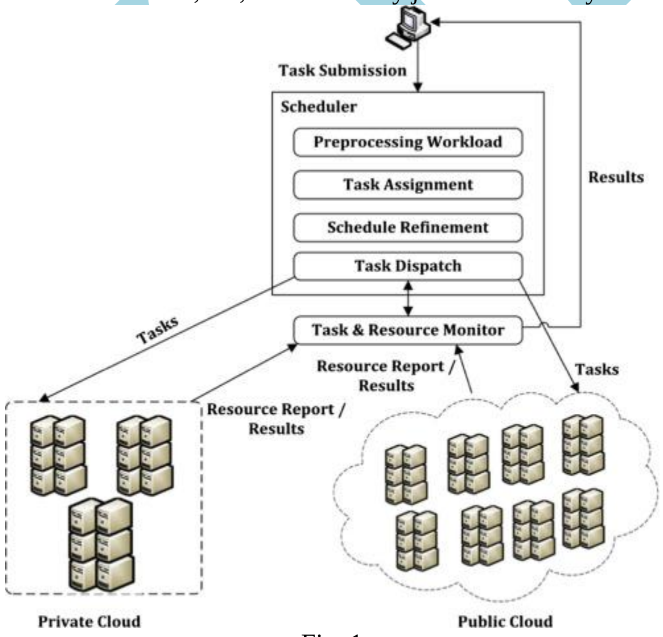

Fig. 1

These dynamic and uncertain factors will lead to a series of problems. On one hand, if the load is too light, the result is a waste of resources; on the other hand, if theload is too heavy, it affects the performance of services running on a node. Therefore, it is important to evaluate and manage cloud resources and their load, in real time. In this paper, a node refers to a server. Moreover, energy consumption is a key problem in cloud computing. Google alone consumed more than 2.3 billion kW/h in one year, and such consumption increases every year. It is therefore urgent to explore effective methods to save energy. Meanwhile, low resource utilization is also a big problem in cloud data centers. The average utilization of a server resource only accounts for between 5 % and 40 % of the total server capacity, i.e. sixty percent of resources are in an idle state [3].

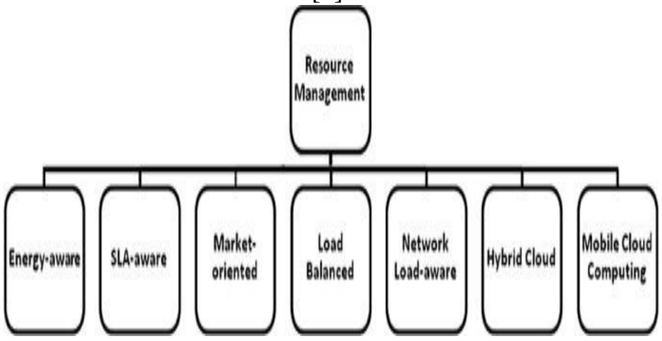

Fig. 2

Therefore, the present solution‘s most typical energy conservation strategy is to turn off these idle resources. A statistical report coming from the High Performance Computing Center of Kyoto University shows that this method saved about 39 % energy by shutting off idle nodes irregularly [4]. Therefore, it is essential to evaluate resource load states and close idle resources in the cloud data center. However, this process will involve VM migration when closing idle nodes. This paper focuses on the following: evaluating the resource load state more accurately in real-time and quantifying the related energy. There are some studies on evaluating resource loads. However, most of them use physical indicators, such as a CPU, bandwidth, memory or other information [5–7]. These static indicators cannot meet our needs because of the difficulty in reflecting the dynamic changes of cloud resources. Some studies have divided the resource load status using a fixed threshold when evaluating the resource load, but such fixed thresholds are unsuitable for dynamic load environments.

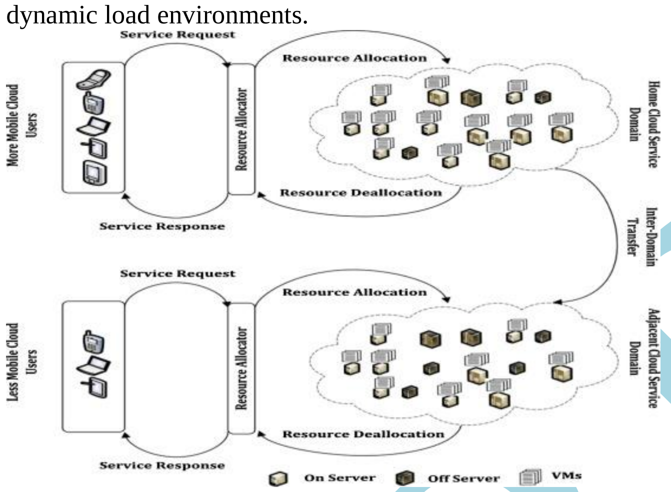

Fig. 3

For example, there may be a variety of applications sharing the same physical resources, and those resources may join or exit at any time. These fixed thresholds cannot meet the dynamic needs of cloud resources and accurately reflect the state of the resource. Some works (e.g., [8]) leveraged the extended form of these indicators, such as CPU utilization. But it is inaccurate that the CPU utilization cannot truly reflect the actual CPU usage state because it also includes memory latency. Additionally, there are some new studies solving this dynamic problem using task scheduling methods, such as [9, 10]. But the function of these methods is limited only by task scheduling and do not solve the energy conservation problem. To mitigate the situation discussed above, this paper evaluated the resource load states using dynamic indicators, and proposes a dynamic self-adaptive threshold to divide the resource load. In particular, this paper also proposes a Selfadaptive threshold based Dynamically Weighted load evaluation Method (termed SDWM). It evaluates the resource state by leveraging three dynamic indicators using the dynamic weighted method, and then divides the resource load into three states using dual thresholds, which include Overload, Normal and Idle. It then releases the idle resources and migrates the overload resources to a balanced load and system improvement utilization rate, proposing a quantifiable energy evaluation algorithm based on the evaluation results. These thresholds used to evaluate the resource states and energy evaluation algorithms are all dynamic and self-adaptive. The contributions are as follows. – First, this study leverages three dynamic indicators (the number of resource requests, resource service capacity, resource service strength) to evaluate the resource usage. These dynamic indicators could more accurately describe the resource state. – Second, this study proposes a dynamic weighted evaluation method to accurately evaluate the resource load states [17]. The method divides the resource load into three states using dual self-adaptive thresholds, which include Overload, Normal and Idle. Then it releases the idle resources and migrates overload resources to balance loads and improve system utilization. – Third, this study proposes a quantifiable energy evaluation model based on the evaluation results [18]. The parameters were obtained using linear regression according to the actual experimental environment. The model can describe the resource energy (including virtual resources) quantitatively and save energy effectively. The rest of the paper is organized as follows. The related work is introduced in Section 2, and the problem description and system model are shown in Section 3. The SDWM is illustrated in Section 4. The energy evaluation algorithm based on the evaluation results will be detailed in Section 5. The analysis and discussion of algorithm is in Section 6. The experiment for SDWM is shown in Section 7, and the paper is concluded in Section 8. .

II. PROBLEMS AND DESCRIPTION

In solving the problem of RAS in cloud computing, researchers have performed significant numbers of studies from various aspects. We can sum up five hot topics of cloud RAS from these researches: (1) localityaware task scheduling; (2) reliability-aware scheduling; (3) energy-aware RAS; (4) SaaS layer RAS; and (5) workflow scheduling. In this section, we discuss the five topics in detail. Problem 1: Locality-aware task scheduling problem how to exploit data locality in RAS of cloud to improve execution efficiency and save network bandwidth. In cloud computing, one of Hadoop's basic principles is "moving computation is cheaper than moving data" [45], which indicates that migrating computation closer to the data source is better than moving data to where the application is run [19]. Furthermore, network bandwidth is a scarce resource, so in order to minimize network congestion and increase the overall throughput of systems, it is necessary to research the problem of locality-aware task scheduling to enhance data locality between jobs. In general, good data locality means that a computation task is located near the data source. There are many researchers [414] attempting to achieve good data locality as much as possible. Problem 2: Reliability-aware scheduling problem how to reduce failure rate of tasks in RAS of cloud, to improve the reliability and execution efficiency of cloud system. In the cloud environment with thousands of data nodes, resource failures are inevitable, which may lead to execution abort, data corruption and loss, performance degradation, service level agreement (SLA) violation, and consequently devastating loss of customers [20]. Therefore, researching the problem of reliability-aware scheduling to improve the reliability of a cloud computing environment is a crucial challenge. For example, MapReduce [46] can automatically handle failures: if a node crashes, MapReduce can re-run its tasks on another node. There are some research works [1518] that consider reliability. Problem 3: Energy-aware RAS problem how to reduce energy consumption of data centres to minimize operating costs of cloud providers. With the demand for high-performance computing infrastructures growing dramatically, the energy consumption of data centres is increasing and has therefore become an important research issue. High energy consumption not only contributes to high operational costs, which will reduce cloud providers' profit, but also emits substantial carbon dioxide (CO2) which is unfriendly to the environment. Therefore, there is an urgent need to research the problem of energy-aware RAS to reduce energy consumption, while maintaining high SLAs between users and cloud platform providers. Many studies [1930] have been performed to find energy-efficient RAS approaches to save energy and reduce the usage of power in cloud computing. Problem 4: SaaS layer RAS problem how to optimize RAS in SaaS layer to maximize profits of SaaS providers. Services in the cloud can be categorized as: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and SaaS [47]. SaaS is a software delivery model providing software service to customers at a low cost across the Internet independently without supplying the underlying hosting infrastructure [48]. SaaS providers obtain profits from the margin between the operational cost of infrastructure and the revenue generated from customers. Therefore, researching the problem of SaaS layer RAS to minimize the overall infrastructure cost without adverse influence to the customer for SaaS providers has become an urgent problem to be solved in the cloud computing environment. There are some other research works [3134] that focus on maximizing the profits of SaaS providers. Problem 5: Workflow scheduling problem how to optimize workflow scheduling to trade off time and cost. Workflows constitute a common model for describing a wide range of applications in distributed systems [49]. Their scheduling methods try to minimize the execution time (makespan) of the workflows in distributed systems like Grids. However, in clouds, there is another important parameter than execution time, that is, economic cost [21]. In general, the faster resources are, the more they cost. Therefore, researching the problem of workflow scheduling to select timecost tradeoffs is a significant project. A workflow application is commonly denoted by a directed acyclic graph, G ¼ (T, E), where T is a set of n tasks (t1, t2, ... , tn), and E is a set of dependencies between tasks. Each dependency, ei,j ¼ (ti, tj) 2 E, where ti is the parent task of tj and tj is the child task of ti, represents a precedence constraint, which indicates that a child task cannot be executed until all of its parent tasks have been completed. Labels on tasks represent computation costs, for example, number of instructions; labels on dependencies represent communication cost, for example, bytes to transmit. An example of workflow is shown in Figure 1, and the main objective of the workflow scheduler is to decide where each task component of the directed acyclic graph will execute.

III. REVIEW LITERATURE

Nasr et al. presented CR-AC using chemical reaction optimization (PSO) and ACO for scheduling problem to reduce cost, time, and complexity [6]. Zhou et al. presented, fuzzy dominance sort based heterogeneous earliest-finish-time (FDHEFT) for workflow scheduling to minimize cost and makespan [7]. Zhou et al. presented, fuzzy dominance sort based heterogeneous earliest-finish-time (FDHEFT) for workflow scheduling to minimize cost and makespan [8]. Micota et al. presented Constraint Programming and Satisfiability Modulo Theory to provide potential lowcost services [9]. Meshkati et al. presented a hybrid scheduling framework based on ABC&PSO to reduce energy consumption [10]. Singh et al presented resource provision and scheduling algorithm using k-means clustering that performed cost, time, and energy reduction [11]. Kong et al. presented an auction mechanism based resource scheduling algorithm, it improves resource utilization [12]. Duan et al. presented a scheduling algorithm using ACO, reduces energy consumption [13]. Ferdaus et al. presented a scheduling algorithm using ACO to reduce energy consumption and cost [14]. Swathi et al. presented a hash-proof technique to secure users' data from leakage attacks [15].

IV. SYSTEM MODEL

This chapter introduces the system model to verify that our research is useful in a real environment. First, the system architecture is introduced in Part A of this section. The thermal model of the CPU and the possibility of thermal balance in a thermal emergency in presented in Part B. The VM live migration is presented at the end of this section.

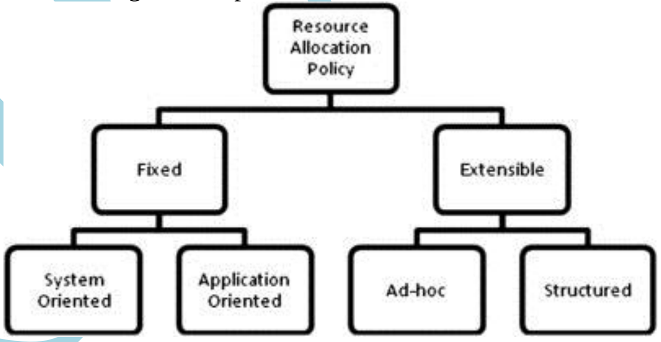

Fig. 4

System Architecture

Figure 2 shows the system architecture. A large number of PMs are connected to each other in an area network in a cloud computing center. Each PM has its own resource pool that provides the usual system resources, such as CPU, memory, and network bandwidth. The VM hypervisor layer (virtual machine manager) is located above the resource pool layer, and manages several VMs and accesses the resources of the resource pool. The storage of the PM is located in a disk array (NAS or NFS); VMs access the storage and checkpoints of the disk array through an area network. The TAVMM operates on a PM in a cloud computing center; it can acquire information on the PM through the area network.

V. PROPOSED EFFICIENT RESOURCE SCHEDULING ALGORITHM

In this section, the core work of proposed resource scheduling has been presented. The scheduling process consists of a systematic flow for identifying work-load requirements before dispatching a resource. The discovery and scheduling process of the proposed technique represented in figure 1. User tasks are represented as an unpredictable workload sent by cloud users. The virtual resource contains VMs resources to evaluate the user demand before scheduling to physical VMs. Available VMs represented the availability of resources. Search process contains the parameters for generating probable solutions. The probable solutions are generated and stored in a feasible solution list. Solution generation process over on max iteration value (100) reached. Solution evaluation contains the objective function to find minimum execution cost. and time.

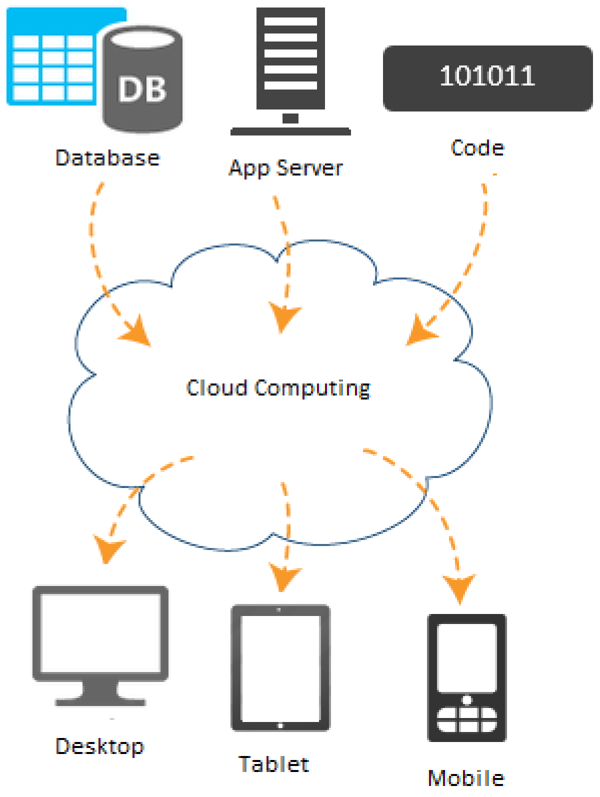

Fig 5

It followed pheromone-based heuristic information to reduce cost and time. Solutions contain minimum execution has selected by sorting order. It inspired by Ants' distributed and intercommunication using pheromone value to find the shortest cost and time effective route. This mechanism helps us to evaluate the solution, while unsuitable solutions sent to evaporation list. Further, a selected solution has sent to the optimal solution that contains the most desirable match. Then an optimal solution has forwarded to resource mapping. Resource mapping performed the function for mapping tasks on VMs. Resource dispatching dispatched VMs based on mapping information. Further, resource scheduling has performed based on dispatching resources. Finally, the task execution process is started based on scheduling decisions, and update information has sent to a resource pool. To manage such an optimization problem, the recommended algorithm has adopted ant mechanism for appropriate matching using the convergence process. The evaluation process is based on ACO parameters, to interact with user tasks and measured the probable solutions. Further, solutions are shortlisted based on heuristic information (intercommunication between solutions) and scheduling has done on a particular VM based on the optimal solution with an objective to minimize execution time. The optimization process scans the resource to prepare a list of the available resource. The search process starts after preparing the list of resource availability, represented by algorithm 1 as shown in figure 2. The tasklist is processing for evaluation until not null. The feasible solution generation process continues until max move criteria not met. The QoS check is a core module of proposed, the feasible solutions made in the above module (algorithm 1) have evaluated in this section, represented in figure 3(algorithm 2). It contains the objective function for evaluating the feasible solution. The objective function designed to achieve a cost-effective solution among solution called optimal solution. The unnecessary solutions sent to evaporation list to save wastage of energy. Objective Function To formulate the scheduling problem, the objective function is used to identify minimum cost and time.

VI. Result and Discussion

In this research work, the objective is to select a suitable VM by improving resource availability (VMs). To evaluate the performance of proposed designed technique cloud-based simulated environment WorkflowSim was used. The experimental results of projected technique have been compared with three existing scheduling algorithms, two heuristic methods: MinMin, MCT (without swarm) and one metaheuristic method ACO (with swarm), in terms of QoS parameters. The scheduling algorithms performance has been tested by running different numerous VMs (5-50) for executing 1000 cloudlets. The algorithm performance has been measured based on execution cost and time of cloudlets (tasks). The results are plotted against execution time; here R2 value is calculated using exponential curve. The R2 value 1 indicates exponential growth and 0 shows an exponential decrease. So, the number of VMs changed, impact on performance, as shown in figures 4-5. The graphical result has shown execution cost and time with respect to VMs. The table 1-2 shows, R2 value of execution cost and time for 1000 tasks. R2 value has proved that SECURE results are more reliable and stable as compared to other policies. It can be deciphered that SECURE performed better compared to other scheduling algorithms in terms of execution cost and time.

REFERENCES

[1]. Kaur S., Bagga P., Hans R.,Kaur H, "Quality of Service (QoS) Aware Workflow Scheduling (WFS) in Cloud Computing: A Systematic Review," Arab J Sci Eng (2015). https://doi.org/10.1007/s13369-018-3614-3

[2]. Mustafa S, Nazir B, Hayat A, Khan A. R., Madni S., "Resource management in cloud computing: Taxonomy, prospects, and challenges," ELSEVIER Comp and Elec. Engg. 47, 186–203, (2015).

[3]. Alboaneen D. A, Tianfield H, and Zhang Y, "Metaheuristic Approaches to Virtual Machine Placement in Cloud Computing: A Review," In: proceeding 15th IEEE Int'l Symp. parallel and distributed comp.,1-8, (2016).

[4]. Singh H., Bhasin A., "Efficient Resource Management Technique for Performance Improvement in Cloud Computing," Indian Journal of Comp. Sci. & Engg. 8(1),33-39, (2017).

[5]. Singh H., Bhasin A., Kaveri P, "QoS based Efficient Resource Allocation and Scheduling in Cloud Computing," Intl. Journal of Tech. and Human Inte (IJTHI), IGI Global, 15 (4), (In Press).

[6]. Nasr, A.A., El-Bahnasawy, N.A., Attiya, G. et al.,"Cost-Effective Algorithm for Workflow Scheduling in Cloud Computing Under Deadline Constraint," Arab J Sci Eng (2018). https://doi.org/10.1007/s13369-018-3664-6.

[7]. Zhou X., Zhang G , Zhang, Sun J., Zhou J, Tongquan Wei, Shiyan Hu, "Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT, Future Genration Comp Sys, 93, 278-289, (2014).

[8]. Zhou X., Zhang G , Zhang, Sun J., Zhou J, Tongquan Wei, Shiyan Hu, "Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT, Future Generation Computer Systems, 93, 278-289, (2014). doi.org/10.1016/j.future.2018.10.046

[9]. Micota F., Eraşcu M., Zaharie D., "Constraint Satisfaction Approaches in Cloud Resource Selection for Component Based Applications," IEEE 14th Intl. Conf. on Intelligent Computer Communication and Processing (ICCP), 443-450,(2018).

[10]. Meshkati, J. & Safi-Esfahani, F.,"Energy-aware resource utilization based on particle swarm optimization and artificial bee colony algorithms in cloud computing," J Super comput, (2010). https://doi.org/10.1007/s11227-018-2626-9

[11]. Singh S, Inderveer C, "Resource provisioning and scheduling in clouds:QoS perspective," Springer Journal of Supercomputing, (2016). DOI 10.1007/s11227-016-1626-x

[12]. Kong Weiwei, Lei Yang, Ma Jing, "Virtual machine resource scheduling algorithm for cloud computing based on auction mechanism," ELSEVIER Optik, 127, pp.5099-5104, (2016).

[13]. Duan H, Chen C, Min G, Wu Y, "Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems," ELSEVIER, Future Generation Computer System, (2016).

[14]. Ferdaus M. H., Murshed M., Calheiros R. N. , Buyya R., "Multi-objective, Decentralized Dynamic Virtual Machine Consolidation using ACO Metaheuristic in Computing Clouds," Wiley Con. and Comp: Prac and exep, 1-40, (2016).

[15]. Swathi G., "Secure data storage in cloud computing to avoiding some cipher text attack," Journal of Information and Optimization Sciences, 39(4), 843-855, (2018).DOI: 10.1080/02522667.2016.1231966

[16]. Chen W, Deelman E, "WorkflowSim: A Toolkit for Simulating Scientific Workflow in Distributed Environments," 8th IEEE Intl. Conf, on eSci 2012, Chicago, (2012).

[17]. S. Dalal & P. Chahar, Deadlock Resolution Techniques: An Overview in International Journal of Scientific and Research Publications (IJSRP), ISSN 2250-3153, Volume 3, Issue 7, July 2013, pp. 1-6.

[18]. A. Saini, S. Dalal and Dr. Kamal Sharma, A Survey on Outlier Detection in WSN, International Journal of Research Aspects of Engineering and Management ISSN: 2348-6627, Vol. 1, Issue 2, June 2014, pp. 69-72

[19]. U. Rani, S. Dalal, and J. Kumar, "Optimizing performance of fuzzy decision support system with multiple parameter dependency for cloud provider evaluation," Int. J. Eng. Technol., vol. 7, no. 1.2, pp. 61–65, 2018

[20]. Mittal, A and Sharma, KK and Dalal, S, Applying clustering algorithm in case retrieval phase of the case-based reasoning, International Journal of Research Aspects of Engineering and Management, vol. 1, no. 2, pp. 14-16, 2014.

[21]. S. Dalal, G. Tanwar, N. Alhawat, "Designing CBR-BDI agent for implementing supply chain system", System, vol. 3, no. 1, pp. 1288-1292, 2013.