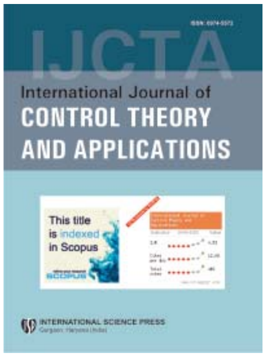

International Journal of Control Theory and Applications

ISSN : 0974–5572

Volume 10 • Number 27 • 2017

Comparison of RECSS Algorithm with EARH and NMEARH

D.S. Misbha[a] and J. R. Jeba[b]

[a]*Assistant Professor, Department of Computer Applications, Nesamony Memorial Christian College, Tamil Nadu, India.*

[b]*Associate Professor and Head, Department of Computer Applications, Noorul Islam Centre for Higher Education, Kanyakumari, Tamil Nadu, India.*

Abstract: Data warehousing in cloud atmosphere is a “growing model” that has developed in such a way that the services based on information technology can be presented. To satisfy the end users’ requirements, data warehousing has altered the method of managing and storing information for compatible, concurrent, applications and resources based on internet. To a greater extent, for cloud services, the remote host machines are built that causes additional capacity indulgence and power utilization. Past few decades, power utilization has turn out to be a significant cost aspect for cloud resources. Because of modern market trends, the evaluation of job allocation strategies in cloud computing must take into consideration both performance and energy utilization. A cycle-precise simulation of the processor and communication activities is too time-consuming, making rapid exploration of the job allocation algorithms become unfeasible. This proposed work presents a CloudSim toolkit that implements the appropriate abstractions for a precise evaluation of the energy consumption of RECSS, EARH AND NMEARH algorithms. A simulation platform called CloudSim toolkit is chosen, in order to compare and evaluate allocation algorithms that discover the agreement between performance and energy. The simulation result shows that the scheduling quality of RECSS is significantly improved compared to others and it is appropriate for real-time job scheduling in virtualized data center in clouds.

Keywords: *RECSS, EARH, NMEARH and Virtualized Clouds.*

1. INTRODUCTION

Due to the powers price intensification, power-associated costs have turn out to be a chief cost-effective factor for IT frameworks and data-centers. Organizations are at present concentrating on the indigence to develop power efficiency. Currently, in the depiction of information and IT resources, the idea of cloud has been a dominating pattern for public and enterprises. The enterprises data-center will necessarily try to exploit its profits by implementing numerous hosted services as feasible, however it is forced by the IT’s infrastructure. The quality of service will humiliate on the reception of several services that leads to unexpected difficulties as per service-level-agreements and also leads to loss in customer satisfaction.

Data warehouse applications are organized in remote data centers (DCs) in which storage systems and high capacity servers are placed. A rapid development of claim for services based on cloud fallout into the establishment of huge data centers that consumes immense amount of power. To lessen functional costs of infrastructure, energy efficient model is necessary that sustains the essential Quality of Service (QoS). Power intensification can be attained by mingling resources based on the present usage and providing well-organized virtual network topologies. Progressively, virtual machines are deemed to be main utilization traces for cloud users. At users end, power management has turn out to be a bottleneck for appropriate functioning of services. A compact between power consumption and performance has become the vital aspect for end users'.

It was identified that providing cloud services, and large-scale data centers results in tremendous amount of power consumption with high cost. Furthermore, due to high power consumption, the system consistency is low. The reason behind the high power consumption in cloud data centers is due to the low utilization of computing resources. As reported by latest studies, in most of the data centers, the average resource utilization (RU) is lower than 30 % [1], and the power consumed by redundant resources is beyond 70 % of peak power [2].

2. RELATED WORKS

Petrucci et al. [3] put forward an approach called dynamic configuration and sketches an efficient algorithm to optimize energy in virtualized servers and dynamically manage the virtualized servers. The preceding work gives a consolidation methodology that was presented by Goiri et al. [4, 5] in order to compact with hesitant information while maximizing performance. New researches [6] were developed to improve QoS in multiprocessing environment by executing Genetic Algorithm (GA) along with Bat Intelligence (BI) that handles energy-aware scheduling problems. This significantly reduces the power consumed by scheduling operation. Jeba et al., in [7] proposed an efficient SMine (Sorted Mine) Algorithm for discovering repeated tasks and to decrease the number of jobs in the list. In SMine [8] algorithm, all repeated itemsets are extracted from the database by limiting the number of searches. In the first search, the number of occurrences of every item is established and the irregular items are rejected. After that, the regular items are tallied in each operation. The operations are arranged depending on the number of regular items in decreasing order. Based on the efficient allocation of computing workload to the resources, in [9], a power-aware job scheduling algorithm was presented on different CPU-GPU architectures. A power-aware DVFS run-time system was presented in [10] that reduce energy with minute performance loss. The scheduling algorithms based on DVFS bring an insignificant reply for workloads with short period since they depend on response rather than prediction and it takes place after the transaction. To surmount the above, in [11], an algorithm was presented in which the volume of performance loss is correlated to the delay between capacity and request and the number of shifts in the workload. Power crest is defined as an approach to diminish peak power under a predestined threshold that is sturdily prejudiced by the jobs triggered into the computing nodes. Thus in [12], a method was proposed to vigorously control the peak power by maintaining the system performance as high as possible.

3. CONTRIBUTION OF THE WORK

A comparative analysis of three scheduling algorithms: RECSS, EARH and NMEARH were presented in this work. A brief description and overview of the three scheduling algorithms are described. A detailed description of the simulation environment and performance evaluation setup is also defined.

4. OVERVIEW OF NMEARH, EARH AND RECSS

In this section, the overview of NMEARH, EARH, RECSS has been explained briefly as follows.

4.1. NMEARH (Non-Migration EARH) Scheduling Algorithm

NMEARH utilizes the rolling-horizon optimization policy to improve schedulability by terminating the jobs with rigid deadline earlier. NMEARH does not employ the virtual machine migration during the allocation of real-time jobs. In addition, NMEARH was deficient in providing a better compact between total energy consumption and assurance ratio.

4.2. Energy-Aware Rolling Horizon Scheduling Algorithm

EARH make use of energy-aware scheduling incorporated with rolling-horizon optimization policy. The virtual machine controller and real time controller are self-possessed in rolling horizon to grasp new job along with the waiting job. Resource scale up and scale down are taken into consideration by the above mentioned algorithm. The result shows improvement in scheduling quality and it saves the energy. The negative aspect is that the maximum number of CPU cycles allocated to a VM must be dynamically updated.

4.3. Rolling Energy Cuckoo Scale Scheduling Algorithm

RECSS is used to minimize the power utilization in the virtualized clouds [15]. RECSS places the whole new and waiting job into a rolling-horizon and the jobs are then scheduled by the cuckoo scale scheduler. The scheduled jobs are permitted to adjust the system schedulability and perhaps minimize power consumption. The energy-aware cuckoo scale scheduling algorithm is in studious fashion. It assigns each job to a VM that forcefully meet job's deadline by preserving power consumption.

5. SIMULATION ENVIRONMENT AND PERFORMANCE SETUP

To evaluate the performance improvement obtained by RECSS, the RECSS algorithm is quantitatively compared with EARH and NMEARH.

5.1. Performance Metrics

The performance metrics estimated in the proposed system comprises the following:

5.1.1. Resource Allocation

Resource Allocation (RA) is defined as the process of conveying available resources to the desired cloud applications. Resource provisioning solves the problem of available resource allocation by permitting the servers to handle the resources for each component efficiently. In order to finish the users' requirement, the type and the number of resources for each application is needed.

5.1.2. Resource Utilization

It refers to a computer's practice of processing resources or the quantity of work supplied by a VM. Resource utilization varies depending on the quantity and type of managed computing jobs. The cloud provider activities for utilizing and allocating limited resources within the cloud surroundings congregate the need of the cloud application.

5.1.3. Execution Time

Execution time is the time required for completing a job. The average time taken to process a job is the total time taken to run a job.

$$\text{Execution Time} = \frac{\text{Million instructions executed on a processor}}{\text{MIPS}}$$

5.1.4. Power Consumption

Power consumption in cloud data centers is a current issue that was considered recently. Many scheduling algorithms were developed to reduce power consumption and to improve performance thus making the cloud services more efficient. The power consumed by a job during execution is defined by

$$P = P_{dynamic} + P_{static}$$
$$= k_1 V^2 M + k_2 (k_1 V^2 M)$$
$$= \alpha V^2 M$$

where, $\alpha \rightarrow$ Proportional constant,

$V \rightarrow$ supply voltage and

$M \rightarrow$ Million instructions executed in a processor.

5.2. Performance Impact of Resource Allocation

This section presents the simulation results of the three algorithms in terms of resource allocation with respect to task count. The task count taken here is in the range of 100. It was observed from the figure 1 that, in RECSS algorithm only a limited amount of resources are allocated compared to EARH and NMEARH. The tasks in RECSS are executed with limited amount of resources. But in EARH and NEARH, the tasks are executed only when the allocated resources is high.

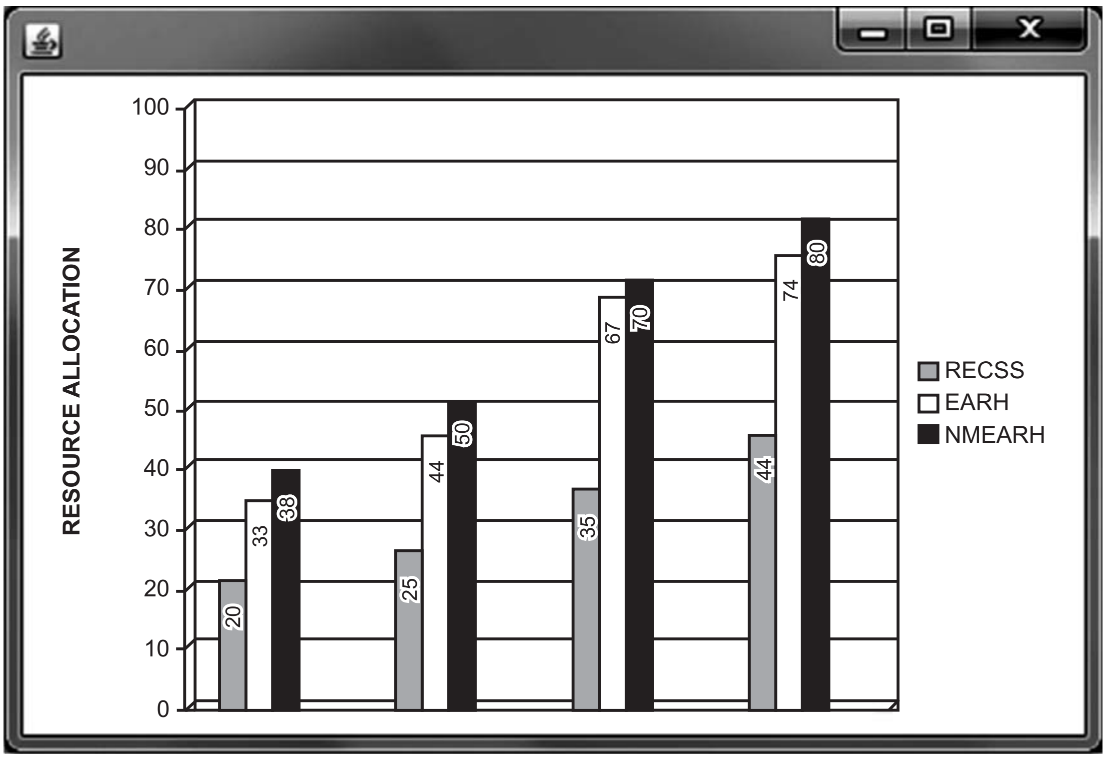

Figure 1: Performance impact of resource allocation

5.3. Performance analysis of Resource Utilization

Figure 2 reveals that the RECSS shows significant improvement in resource utilization compared with EARH and NMEARH. This comparison is accredited by VM migration policy. The VM migration policy makes the system to completely utilize the ability of host computing. RECSS surpasses EARH and NMEARH with 45 and 38 percent, respectively. From this it was clear that RECSS utilizes the resources well to execute the jobs.

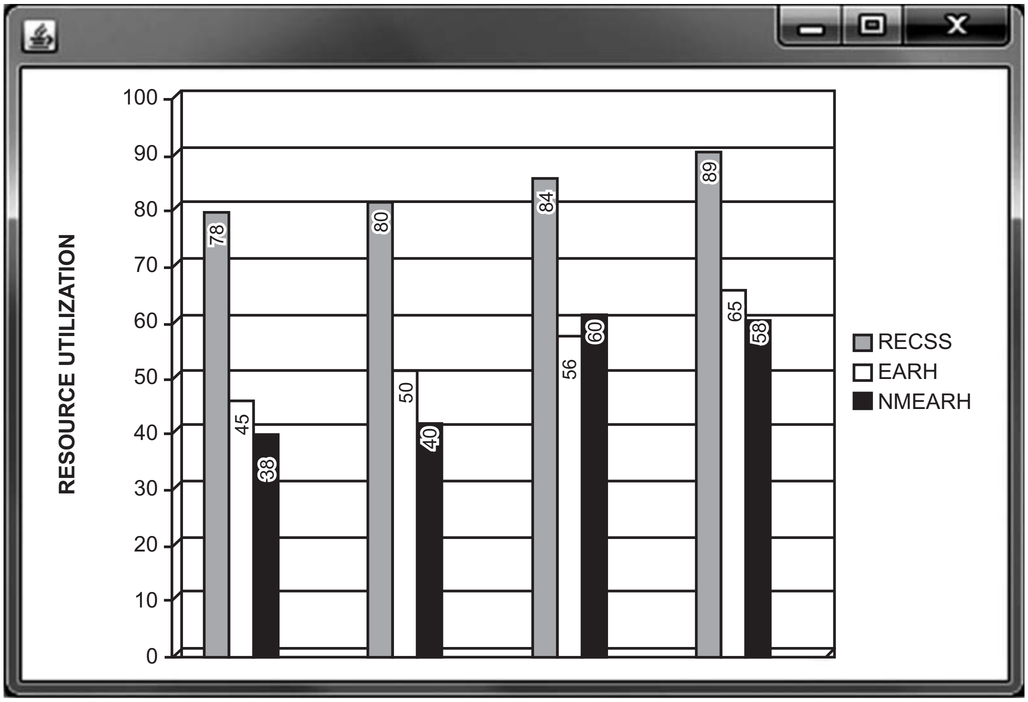

Figure 2: Performance Analysis of resource utilization

5.4. Performance Analysis of Power Consumption

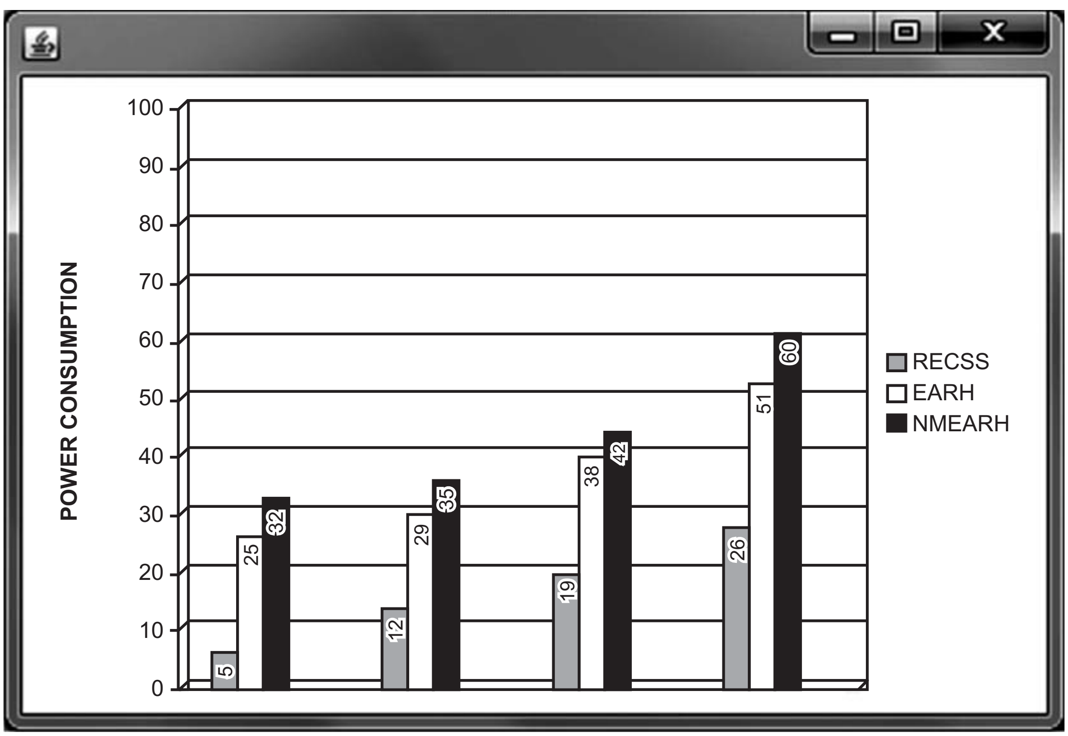

Figure 3: Performance Analysis of Power Consumption

From figure 3, it was observed that RECSS saves more power when compared with EARH and NMEARH. NMEARH and EARH conserve less power. Thus, RECSS has better energy conservation capability in contrast with EARH and NMEARH, and hence the development was more evident with increase in task count. The above observation points out that the utilization of VM migration policy is very efficient while scheduling concurrent jobs. From one point of view, if the task count increases, the current VMs are merged to make some space for creating new VMs, in order to avoid the power consumption caused due to the addition of new energetic hosts. On the other point of view, the VMs in light-load host can be migrated to other hosts which minimize the energy consumption by further turning down the inactive hosts.

5.5. Performance Impact of Execution Time

The execution time is calculated based on the million instructions per second (MIPS). From the Fig.4 it was clear that the execution time is low in the proposed RECSS algorithm than EARH and NMEARH. The execution time is high in NMEARH due to the ignorance of VM migration. The execution time increases with increase in task count which leads to high resource demand. The resource demand decreases sharply with the submission of few tasks into the system.

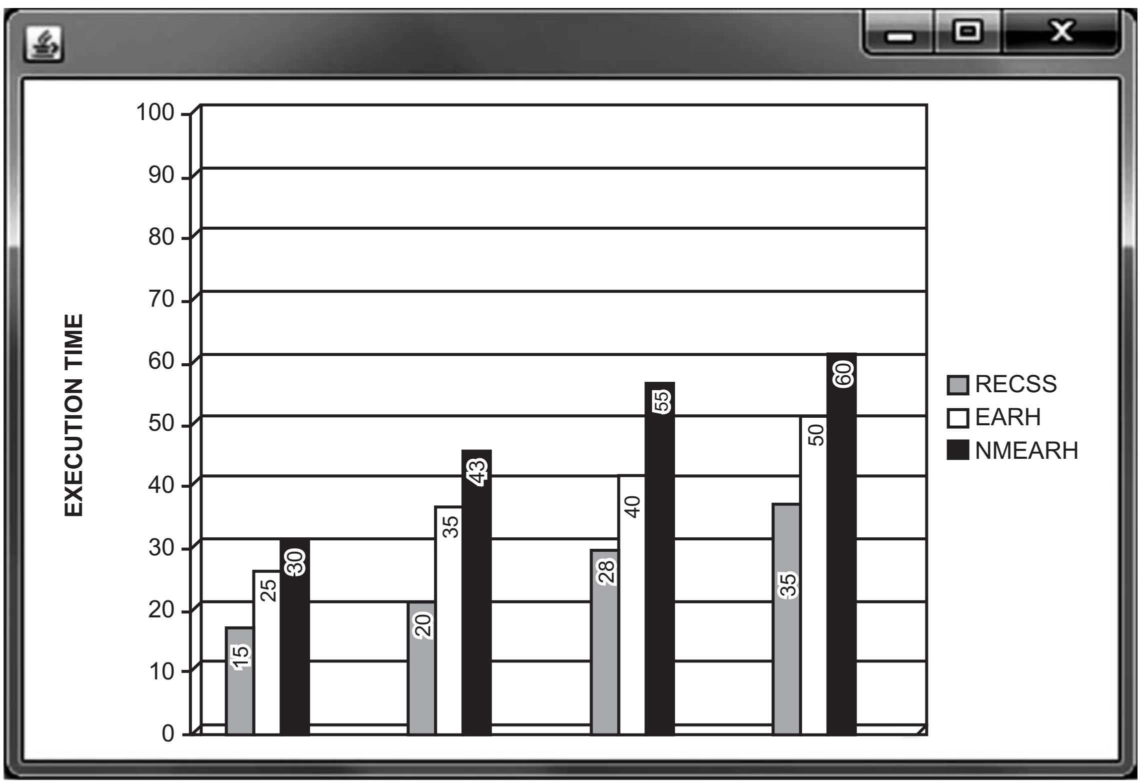

Figure 4: Performance impact of execution time

6. COMPARISON CHART

Table 1 shows the comparison chart of the three algorithms. The three algorithms are compared in terms of performance metrics such as resource allocation, power consumption, resource utilization and execution time for task count 100.

Table 1
Comparison of RECSS, EARH and NMEARH with performance metrics.

Performance Metrics	*RECSS*	*EARH*	*NMEARH*
Resource Allocation (%)	20	33	39
Power Consumption (*106) W	5.23	25	32.2
Resource Utilization (%)	78	45	38
Execution Time (sec)	15	25	30

7. CONCLUSION AND FUTURE SCOPE

This paper presents a comparative analysis of RECSS with various scheduling algorithms such as CS, EARH and NMEARH, taking into consideration the energy awareness for optimal performance of cloud data centers and achieves excellent provisioning of resources. The algorithms focus on various parameters such as power consumption, resource allocation, resource utilization, and execution time. The evaluation shows that the proposed RECSS algorithm outperforms significantly the other three algorithms. The future work will focus on the extension of the proposed work and to utilize the extended work in a real-cloud environment to test its performance.

REFERENCES

[1] L.A. Barroso and U. Hlzle, “The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines,” Synthesis Lectures on Computer Architecture, vol. 4, no. 1, pp. 1-108, 2009.

[2] X. Fan, W.D. Weber, and L.A. Barroso, “Power Provisioning for a Warehouse-Sized Computer,” ACM SIGARCH Computer Architecture News, vol. 35, no. 2, pp. 13-23, 2007.

[3] V. Petrucci, O. Loques, and D. Moss´e. A Dynamic Configuration Model for Power-efficient Virtualized Server Clusters. In 11th Brazillian Workshop on Real-Time and Embedded Systems (WTR), 2009.

[4] J. Berral, ´I. Goiri, R. Nou, F. Juli`a, J. Guitart, R. Gavalda, and J. Torres. Towards energy-aware scheduling in data centers using machine learning. In 1st International Conference on Energy-Efficient Computing and Networking (eEnergy’10), pages 215–224, 2010.

[5] ´I. Goiri, F. Juli`a, R. Nou, J. Berral, J. Guitart, and J. Torres. Energy-aware Scheduling in Virtualized Datacenters. In 12th IEEE International Conference on Cluster Computing (Cluster 2010), 2010.

[6] C. -t. Ying and Y. Jiong, “Energy-aware Genetic Algorithms for Task Scheduling in Cloud Computing”, 2012 Seventh China Grid Annual Conference, (2012) September 20-23, Beijing, pp. 43-48, IEEE.

[7] Dr. J.R.Jeba & Dr.S.P.Victor, “A Novel approach for finding Frequent Item Sets with Hybrid strategies,” International Journal Of Computer Applications, Vol. No: 17, No: 5, Pg : 30-33.

[8] Dr. J.R.Jeba & 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.

[9] Matteo Chiesi, Luca Vanzolini, Claudio Mucci, Eleonora Franchi Scarselli, and Roberto Guerrieri, “Power-aware job scheduling on heterogeneous multicore architectures,” IEEE Transactions on Parallel and Distributed Systems, Vol. 26, No. 3, March 2015.

[10] C.-H. Hsu and W.-C. Feng, “A power-aware run-time system for high-performance computing,” in Proc. ACM/IEEE Conf. Supercomput., 2005, p. 1.

[11] W. L. Bircher and L. K. John, "Core-level activity prediction for multicore power management," IEEE J. Emerging Select. Topics Circuits Syst., vol. 1, no. 3, pp. 218–227, Sep. 2011.

[12] X. Wang, M. Chen, C. Lefurgy, and T. W. Keller, "SHIP: A scalable hierarchical power control architecture for large-scale data centers," IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 1, pp. 168–176, Jan. 2012.

[13] Amir Vahid Dastjerdi , Sayed Gholam Hassan Tabatabaei, A dependency-aware ontology-based approach for deploying service level agreement monitoring services in Cloud, Journal Volume42,April2012 Pages 501-518.

[14] Sheheryar Malik, Fabrice Huet, "Adaptive Fault Tolerance in Real Time Cloud Computing," services, pp.280-287, 2011 IEEE World Congress on Services, 2011.

[15] D. S. Misbha and Dr. J. R. Jeba, "Scheduling effective Cloud Updates in Streaming Data Warehouses using RECSS Algorithm," International Journal of Applied Engineering Research, Vol. 11, ISSN 0973-4562, No. 5, 2016, pp.3632-3636.