Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 3, June 2021:50 - 61

Research Article

Deadlock Preventıon Usıng Schedulabılıty In Vırtualızed Cloud Envıronment

K.Saranya[1], R.Ramya[2], S.Revathi[3], N.Gomathi[4]

Abstract

Cloud computing is to provide a on demand IT resources/services like server, storage, database, networking, analytics, software etc. over internet. It provide a diversity of distributed computing system, and it has a group of virtualized and inter-connected systems based on provider–stage agreements between the consumers and service providers. Here, a computer architecture and algorithms were developed for cloud platform via the dynamic provisioned of virtual systems to perform and support soft and real -time task scheduling. The architecture integrated soft real-time task scheduling algorithms, namely master node and virtual machine node schedules. In addition Adaptive Adaptive Scoring Job scheduling algorithm is also applied. The motive of cloud computing is to tie together the power of both distributed computing and parallel computing as well as aggregate idle resources on the Internet such as Central Processing Unit (CPU) cycles and storage spaces for better consumption. Cloud computing, which connects a commodity hardware with high speed networks, that can meet the same computing power as a supercomputer does, with a lower cost. However, cloud is a heterogeneous system. Independent tasks scheduling in cloud is more complicated. To make use of the power of cloud entirely, we require an well-organized job scheduling algorithm to allocate jobs to resources in a cloud. This project proposes an Adaptive Scoring Job scheduling algorithm (ASJS) for the cloud environment. Compared to other algorithms, it can reduce the time of job completion, which may compose of computing-intensive jobs and data-intensive jobs. Python 3.6 is used as the front end language to develop the application.

Keywords: *Soft and Real -time task scheduling, cloud computing, virtual machine, Adaptive Scoring Job Scheduling algorithm*

[1,3,4]Assistant Professor, Dept. of CSE, Jai Shriram Engineering College, Tirupur, Tamilnadu, India.,
[1]saranya@jayshriram.edu.in ,[3]revathi@jayshriram.edu.in , [4] gomathi@jayshriram.edu.in
[2] P.G Scholar, Dept. of CSE, Jai Shriram Engineering College, Tirupur, Tamilnadu, India
[2]ramyacs024@gmail.com
Received Accepted

Introduction

In over the last few years, the demand for cloud computing is increased, because of the aggressive growth of big data generated by the social media and other resources. Cloud computing offers a new mode of service provision by reorganizing various resources over the internet[23]. Computing architecture and algorithms has been developed to support soft real time task scheduling. There are three customized soft real time task scheduling algorithms are integrate in this architecture. To maintain system criticality, deadline look-ahead module was integrated to find out that exception, and missing deadlines in the system. Software as a Service (SaaS) , Platform as a Service(PaaS) and Infrastructure as a Service (IaaS) are the service models of cloud environment. In this paper, the chance of applying real-time task scheduling for deadline controlled task scheduling has been analysed. Multi agent gaming, online video streaming, and telecommunication Management are the examples of such applications. It could be take much advantage from Cloud Computing. With the ability of cloud computing, elastic allocations of resources are thereby enabled in order to support the dynamic workloads.

In this work concerned about the problem of scheduling independent period in real time task with implicit deadlines on virtualized cloud environment . A periodic real time task is released at regular intervals in real time system and it can be found in a wide range of real time system applications. Monitoring the growth of plant or air travel control systems are some of the examples of such applications. Altering the track in a radar system over a fixed interval is an another example. Polling information from sensor is an additional example. The constant intervals may be varied from system to system in real time computing task. The model of the architecture have been summarized in following.

In hard real-time computing system, G C Buttazzo [2] proposed predictable scheduling algorithms and applications. In this edition, an necessary description on real-time computing has been updated. Here it shows the development of fundamental concepts in real time computing. Then, it also provides the necessary methods to design predictable computing systems for the field. Previous edition contains the assessment of latest developments in Real Time systems, which includes resources reservation techniques, overload handling algorithms, Limited pre-emptive scheduling and adaptive scheduling techniques. Each chapter provides the basic definitions of each concepts and algorithms and examples with proper figures.

There are two type of Real Time Systems such as hard real time and soft real time systems. By proposing the low relative communication level, the delay caused by the communication between Virtual Machine nodes and Master nodes are considered to be zero, and which are also supporting high speed networks. There are so many algorithms are developed for cloud environments. In this, the most of the algorithms are mainly targeted to Resource Management [16] to [19] performance cost-effectiveness [21] and energy conservation [18][22]. Few algorithms are proposed on deadline based scheduling in cloud environment [16].

Lıterature Revıew

The researches made on to find out the solution of load-balancing ,are statically focused on constant scenarios, which means, in most of the cases its like employ heuristic methods. On overall view of last few years, an easy adaptable search technique named genetic algorithm has highly popularized because of its robustness. The work assigned here is to sort out the dynamic load-balancing exceptions and to find out the procedure of how a genetic algorithm has been raised to solve the problem. A dynamic load-balancing algorithm is commonly developed for optimal or near-optimal task allocations, because it can evolve during the operation in the parallel computing system. [1]

The authors of this paper [2] has exposed about Fair queuing technique. It is a technique which permits each flow to pass through a network device to adopt a fair share network resources. The previous schemes of fair queuing techniques were nearly achieved perfect fairness.And it costs expensive to implement: specifically, on work requirements. To process a packet under these schemes , it was taken O(log(n)), where n is the number of active flows. This will leads to the level of becoming expensive at high speeds. On the other side , the cheaper approximations and calculations of fair queuing technique have been reported in the literature survey for its unfair behaviour. So , the authors involves in a process of describing a new approximation of fair queuing technique , and so it is called Deficit Round Robin. This scheme nearly achieves a perfect fairness level in terms of throughput, and it requires only one (O(1)) work to process a packet, and is more enough to implement this process in hardware. Deficit Round Robin is also capable to solve other scheduling exceptions which arise in between in the area of servicing and which cannot be split up into smaller units, for the distribution in queues.

The cloud service providers incorporates the highly heterogeneous, and multi-vendor interconnection of devices to allow multi qualitative gradations of services. During each phase of lifecycle started from data placement to retirement of workload, it remains the same monumental task for the cloud providers. It is used to perform the following process: planning, configuration and migration on demand. Such operations are always called free of errors, and it has the complexity of interrelated, physical and logical provisioning. Cloud providers have chosen a variety of different approaches to prevent the effects from scalable storage. [3]

Exıstıng System

In existing system, a deadline look-ahead module was implemented using EDF Algorithm to identify deadline exceptions and to keep away from missing deadlines. It develops a computing algorithms and architecture in a cloud computing environment for supporting soft real-time task scheduling through the dynamic provisioning of virtual machines. The result is in addition to periodic real-time tasks of the soft real-time scheduling in cloud computing, it also handle the near-hard real-time task scheduling.

Drawbacks of existing system:

- Each task is considered as a separate unit.
- The entire task is given to a single virtual machine
- Splitting of single tasks into subtask and assigning into various Virtual machines are not considered.

Proposed System

In proposed system Adaptive Scoring Job Scheduling algorithm (ASJS) for Job execution is being carried out along with existing system implementation, In addition to the existing system implementation, to split the task to available computer resources , jobs can be divided into equal sub tasks and provide to one or more clusters. Then Cluster Score is calculated. Storage capacity requirement is also concerned in Cluster Score calculation. During task assignment to VMs, to fit the tasks with required operating system in the VM with corresponding operating systems (either windows or linux) is also carried out. Also, time factor is considered for task execution success or failure.

The cluster score value is calculated using the following formula.

$$\mathbf{CS_i = ATP_i + ACP_i}$$

ATPi means the average available bandwidth the cluster i can supply to the job j and is defined as:

$$ATP_i = \frac{\sum_{j=1}^{m} Bandwidth\_available_{ij}}{m-1}, \quad i \neq j$$

Similarly, ACPi means the average available CPU power cluster i can supply to the job k and is defined as:

$$ACP_i = \frac{\sum_{k=1}^{n} CPU\_Speed_k * (1 - load_k)}{n}$$

Advantages of the proposed system :

- Cluster score values are always been recalculated while the job is partially ended. When the given sub task is completed, this will be achieved.
- Job Split method assists in faster job completion.
- Storage capacity of cluster resources is also in consideration.
- Effective utilization of virtual resources

System Architecture

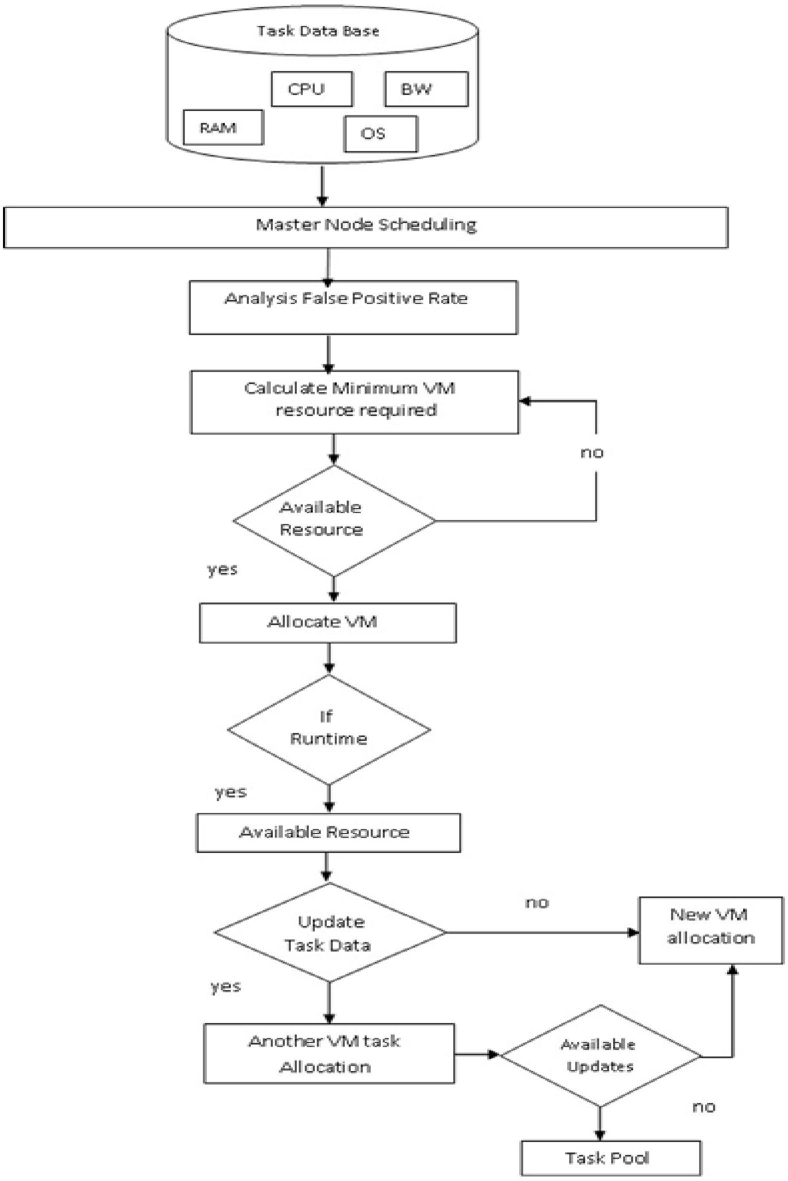

Fig.1 System Architecture Diagram

Implementatıon

a. Tasks details collection

In this module, task details such as required RAM (in MB), Storage Capacity (in MB), and CPU required (in MHz) and network bandwidth (in Mbps) are retrieved and stored in various dictionary entries.

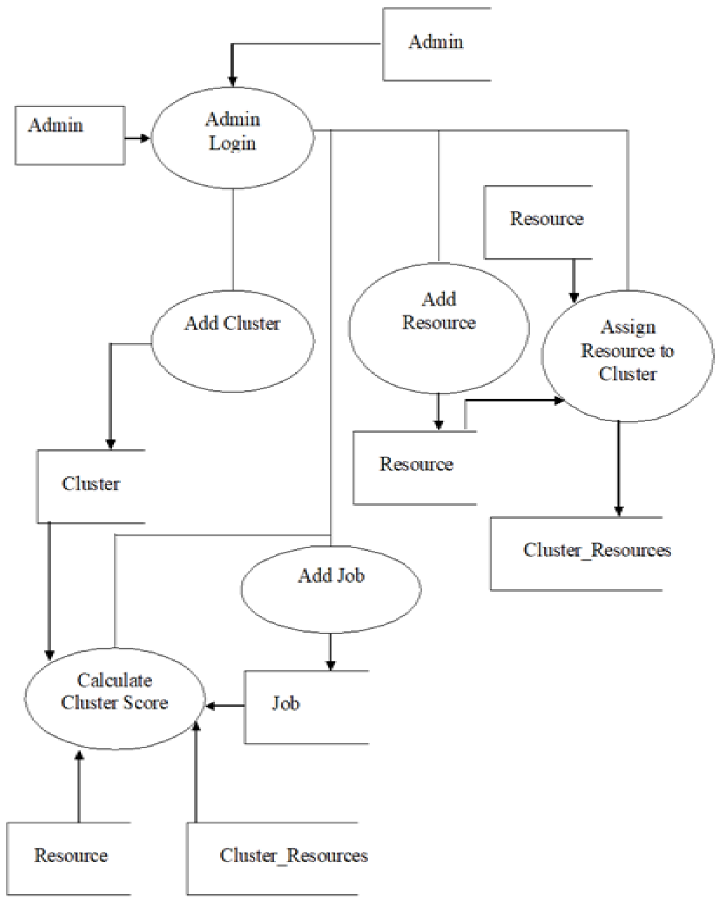

Fig.2 Data Flow Diagram

b. Execution of MN Scheduler, VM Node Scheduler and Deadline Exception Handler

In this module, the algorithms mentioned in system methodology part are executed so that creation of new virtual machine is required or not is found out. Virtual machines' pending resources are updated each time a task is taken and processed.

c. Add Cluster

In this module, the cluster id is given with Average Transmission Power and Average Computing Power and Cluster Score value set to zero. These informations are stored in Cluster matrix.

d. Add Resource

In this module, the ID, Name, IPAddress, CPU MHz, CPU MHz available, Load Percent, Available computing power and Storage Capacity of resource are keyed in. The above informations stored in 'Resources' matrix.

e. Assign Cluster to Resource

In this module, the cluster id is fetched from 'Clusters' matrix and resource id is fetched from 'Resource' matrix. The ID' s are selected from combo boxes and are stored in 'Clusters Resources' matrix.

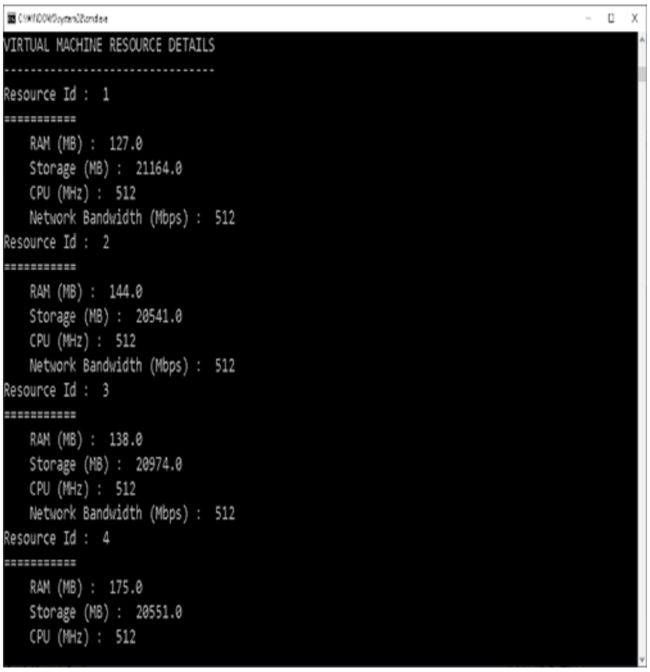

Fig.3 Virtual Machine Resource Allocation Process

f. Add Job

In this module, the Job ID, Name, required RAM capacity, required hard disk storage in MB, Central Processing Units and network bandwidth is keyed in and stored into 'Jobs' matrix

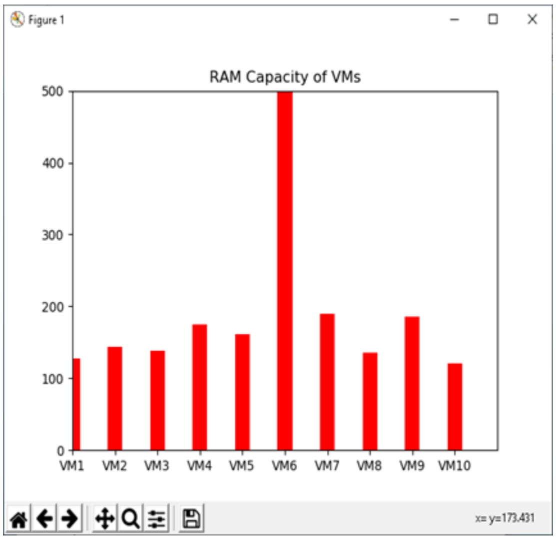

Fig.4 Capacity of RAM in Virtual Machines

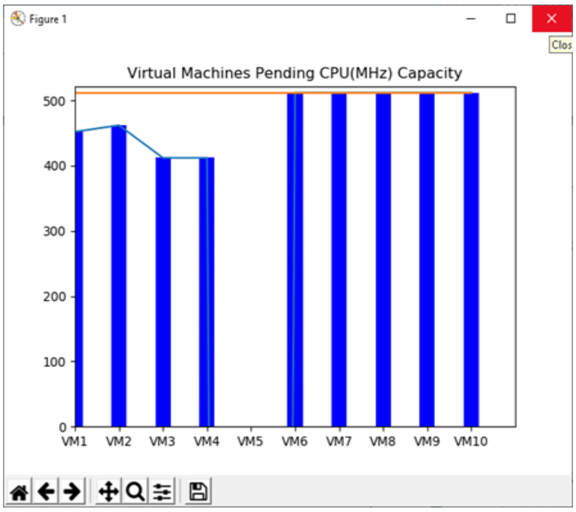

Fig.5 Virtual Machine pending CPU Capacity

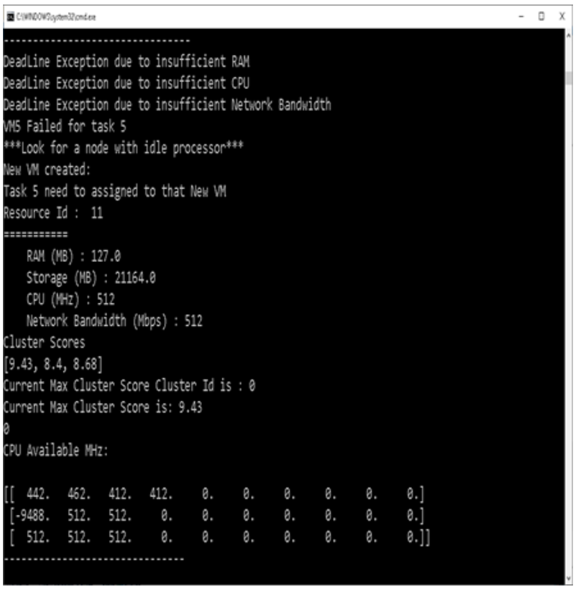

Fig.5 Cluster score calculation with storage capacity.

Conclusion

Here , we developed a architecture and algorithm based on cloud computing for soft and hard real time scheduling. This system is mainly proposed to dynamic allocation and release of computing resources to response of workload needs in cloud environment. To execute the above idea, deadline look ahead schedulers were developed in order to fire the deadline exceptions , where before their occurrence. Here the VM node and MN performs a major role.

VM node detects the exception and sends the causing information about tasks to the MN. Then , MN is suppose to make its attempt to assign the tasks to the available empty processor. If then it will be not available, a new VM node is created dynamically to allot and assign the task.

Reference

[1] A.Burns and A.J.Welllings, Real- Time Systems and programming languages, 4th ed. Toronto, ON ,Canada: Pearson Education,2009.

[2] G .C .Buttazzo ,Hard Real - Time Computing System: Predictable scheduling Algorithms and Application, vol. 24 , 3rd ed. New York , NY , USA : Springer, 2013

[3] J.W. S. Liu, Real – Time System ,1st ed. Upper saddle River , NJ ,USA: Prentice ,2013.

[4] R. Mall, Real – Time System :Theory and Practice .London, U.K : Pearson,2009.

[5] J. A. Stankovic and K. Ramamritham, ‘‘What is predictability for Real-time systems?’’ Real-Time Syst., vol. 2, no. 4, pp. 247–254, 1990.

[6] S. Funk and V. Nanadur, ‘‘LRE-TL: An optimal multiprocessor scheduling algorithm for sporadic task sets,’’ in Proc. 17th Int. Conf. Real-Time Netw. Syst., 2009, pp. 159–168.

[7] G. Levin, S. Funk, C. Sadowski, I. Pye, and S. Brandt, ‘‘DP-FAIR: A simple model for understanding optimal multiprocessor scheduling,’’ in Proc. 22nd Euromicro Conf. Real-Time Syst. (ECRTS), Jul. 2010, pp. 3–13.

[8] B. Andersson and E. Tovar, ‘‘Multiprocessor scheduling with few preemptions,’’ in Proc. 12th IEEE Int.] L. Mao, W. W. Lin, B. Liu, and Y. Da Li, ‘‘An energy-efficient resource scheduling algorithm for cloud computing based on resource equivalence optimization,’’ Int. J. Grid High Perform. Comput., vol. 8, no. 2, pp. 43–57, 2016.

[9] H. Chen, X. Zhu, H. Guo, J. Zhu, X. Qin, and J. Wu, ‘‘Towards energyefficient scheduling for real-time tasks under uncertain cloud computing environment,’’ J. Syst. Softw., vol. 99, pp. 20–35, Jan. 2015.

[10] X. Zhu, L. T. Yang, H. Chen, J. Wang, S. Yin, and X. Liu, ‘‘Real-time tasks oriented energy-aware scheduling in virtualized clouds,’’ IEEE Trans. Cloud Comput., vol. 2, no. 2, pp. 168–180, Apr./Jun. 2014.

[11] M. Dertouzos, ‘‘Control robotics: The procedural control of physical

processes,'' in Proc. Inf. Process., 1974, pp. 807–813.

[12] E. Massa, G. Lima, P. Regnier, G. Levin, and S. Brandt, ''OUTSTANDING PAPER: Optimal and adaptive multiprocessor real-time scheduling: The quasi-partitioning approach,'' in Proc. 26th IEEE Euromicro Conf. RealTime Syst., Jul. 2014, pp. 291–300

[13] U. C. Devi, ''Soft real-time scheduling on multiprocessors,'' Ph.D. thesis, Univ. North Carolina, Chapel Hill, NC, USA, 2006.

[14] J. Lee, A. Easwaran, and I. Shin, ''Laxity dynamics and LLF schedulability analysis on multiprocessor platforms,'' Real-Time Syst., vol. 48, no. 6, pp. 716–749, 2012.

[15] Z.-H. Zhan, X.-F. Liu, Y.-J. Gong, J. Zhang, H. S.-H. Chung, and Y. Li, ''Cloud computing resource scheduling and a survey of its evolutionary approaches,'' ACM Comput. Surveys, vol. 47, pp. 63:1–63:33, Jul. 2015.

[16] J.-T. Tsai, J.-C. Fang, and J.-H. Chou, ''Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm,'' Comput. Oper. Res., vol. 40, pp. 3045–3055, Dec. 2013.

[17] M.-A. Vasile, F. Pop, R.-I. Tutueanu, V. Cristea, and J. Kołodziej, ''Resource-aware hybrid scheduling algorithm in heterogeneous distributed computing,'' Future Gener. Comput. Syst., vol. 51, pp. 61–71, Oct. 2015.

[18] E. N. Alkhanak, S. P. Lee, and S. U. R. Khan, ''Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities,'' Future Gener. Comput. Syst., vol. 50, pp. 3–21, Sep. 2015.

[19] Y.-J. Chiang, Y.-C. Ouyang, and C.-H. Hsu, ''Performance and costeffectiveness analyses for cloud services based on rejected and impatient users,'' IEEE Trans. Services Comput., vol. 9, no. 3, pp. 446–455, May/Jun. 2016.

[20] L. Mao, W. W. Lin, B. Liu, and Y. Da Li, ''An energy-efficient resource scheduling algorithm for cloud computing based on resource equivalence optimization,'' Int. J. Grid High Perform. Comput., vol. 8, no. 2, pp. 43–57, 2016.

[21] Dr.S.Rajalakshmi Ms.K.Saranya "An Efficient Audit Service Outsourcing For Data Integrity in Clouds" International Journal on Engineering Technology and Sciences, Volume 1, Issue 7, Pages: 194-198, 2014.

[22] Ms. N. Gomathi, Ms. Meenalochini M , Ms. K. Saranya "cloud backup services

using genetic algorithm journal", International Journal for Scientific Research & Development, Volume 6, Issue 7, Pages: 336-339, year:2018.

[23] A Saranya, N. Muthuselvi, P , K Saranya "Secure Authorized Deduplication with Data Mining in Cloud", International Journal of Computer Science and Programming Languages", Vol 3, issue-1,2017.