



IOT based visualization of weightage based static task scheduling algorithm in datacenter

Preethi M*, Kayalvizhi Jayavel

^{1,2}Dept. of Information Technology, SRM University, Chennai

*Corresponding Author Email: preethi1010.cse@gmail.com

Abstract

Cloud computing has raised majorly to provide everything as a service and also for scaling the resources and utilizing the resources in an effective way. This paper aims to propose a scheduling algorithm which allocates static tasks to the resources effectively without making any tasks starve for the resources for long time. In SJF algorithm, the shortest tasks will be executed initially, and the largest tasks will keep on starving for the resources to be allocated. The proposed algorithm handles such a situation effectively by adding the jobs under different weightage queues and then scheduling them in an SJF order. This gives priority to even largest job. In this paper a framework is proposed, which fetches data from Amazon SDB storage and the processing of data based on proposed algorithm occurs in a cloudsim and finally the results are visualized through an IOT mobile device. The comparison is also made for First Come First Serve (FCFS), which is a default scheduling algorithm and the proposed algorithm.

Keywords: Cloud Computing, Resource starving, Shortest Job First, Weightage queues, First Come First Serve, Scheduling, Cloudsim, IOT

1. Introduction

Many researchers tried to define cloud computing but still this definition given by U.S. National Institute of Standards and Technology is more appropriate. "The Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resource (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [16]. This definition explains cloud in general; its characteristics and deployment all at one go.

There were two types of tasks submitted to the datacenter broker: static and dynamic. The static tasks are those whose processing time and its entire execution can be pre-determined whereas dynamic tasks are the ones whose size, processing time all can be calculated only at run time.

The biggest challenge in cloud computing is to schedule the tasks and resources. This is because the cloud provider not only deals with single user but with many users. There were many researchers working with the scheduling algorithms. The purpose of this scheduling algorithm of task is to reduce the execution time of task and maximise the resource utilization [18]. This paper aims at providing the good scheduling algorithm which doesn't make any tasks starve for the resources for the long time.

2. Background

An efficient task scheduling strategy should focuses on completing the tasks within the deadline and hence more number of tasks can be allocated to the resources. The algorithm should also suggests reusing of the resources when the allocated tasks get completed. The effectiveness of scheduling algorithm is in reducing the time consumption of processing the tasks and utilizing the resources in an effective way.

Abirami S.P and Shalini Ramanathan mainly focuses on Quality of service. The algorithm considers both the virtual machine resources and the tasks submitted by the users. There were several Quality service parameters. The user request which satisfies most parameter will be allocated with resources [1].

Amdani, S. Y., and S. R. Jadhao proposes a cost-priority algorithm based Multi-Queue methods to increase the throughput and increase the measure of fairness [2].

The efficiency in delivering the services in a cloud majorly depends on how the scheduling of user jobs (tasks) allocated over its resources [3].

Ding, Ding, Xiao Cong Fan, and Siwei Luo says, most of the existing research and practices looks only to increase the profits of cloud providers, thereby resulting in showing minimized attention to the real needs of cloud users. To meet a user's resource needs a relevance feedback network, based resource scheduling mechanism which can be employed by a cloud [4].

Fang Y, in his paper suggests a task scheduling algorithm which improves the resource optimization. This paper proposes a algorithm which takes the available resources in different machines and also collects the tasks submitted by users. The balancer is a component which allocates tasks to the available resources based on the load (number of tasks per second). If the load is minimum, all the tasks will be allocated to a single host, otherwise the tasks will be allocated to different hosts [5].

Flip and Ion-Dornial proposes microservice-oriented architectural approach for scheduling the tasks submitted to the datacenter broker. The heterogeneous environment is setup and the scheduling policy runs over there [6].

Khalil proposes a agent based scheduling algorithm. In his work, the hosts machines which have similar quantity of resources are clustered. There were several clusters formed in this way by following Bayesian network algorithm. Then each cluster is represented by an agent. This agent acts as a datacenter broker in cloudsim and allocates task to the resources available [7].

Li and Zhongjin proposes the methodology which reduces the power consumption and saves energy. The suggested methodology uses the selection algorithm which selects the virtual machines to which tasks to be allocated. The idle virtual machines are made to reuse to effectively utilise the energy and reduce wastage of energy and resources [8].

Madhumathi, Ramasamy proposed a Clustering algorithm along with preference given by user scheduling technique. The four stages of this scheduling algorithm are: matching, selection, clustering and scheduling of resources. The user who puts forward the same requests will be given higher precedence. The VM also chooses the best scheduling algorithm in this proposed algorithm [9].

Cloudsim is chosen for the simulation since cloud user can run and deploy their large scaling applications in a simulated environment and can view their results effectively. It is also easy to manage the large scale resources at a single environment [10].

Salehi and Buyya, in his paper proposes a dynamic resource scheduling algorithm where the execution time of the process or tasks submitted to datacenter broker are unknown. During runtime it calculates the processing time and schedules the tasks [11].

S. Selvarani and G.S. Sadhasivam proposes an algorithm that groups tasks by setting a threshold and allocates those clustered tasks to the appropriate resources [12].

Sharma in his paper proposes the different models of scheduling algorithm and majorly focuses on the managing the workflow of a scheduling algorithm [13].

The main focus of scheduling algorithm is based on improving the effectively both cost-wise and resource-utilization wise. The scheduler should sort the tasks so they can improve the effectiveness of services and also maintains the efficiency among

the tasks. There are two policies which can be defined: Space-Shared scheduling policy and Time-Shared scheduling policy.

In space-shared scheduling policy, it schedules one task on virtual machine at a given point of a time and after its completion it will schedule another task on virtual machine. In Time-Shared scheduling policy it schedule all tasks on virtual machine at the same time. It shared the time among all tasks and schedule simultaneously on the virtual machine. Space shared scheduling policy shows better results as compared to Time-shared scheduling policy [14].

Simão, José, and Luís Veiga proposes Customer driven approach for scheduling tasks based on the Service Level Agreements defined by the customers. The approach brings benefits to customers since it is customer driven and also cost effective [15].

Tsai and Chun-Wei in his proposes a task scheduling algorithm which is prediction based learns the nature of task on its own. This approach detects the best policy to be adopted for each incoming task and allocates task to the resources available [17].

Wen and Zhenyu focuses on framework that maintains the data confidentiality. The framework runs over hybrid cloud platform which combines both private as well as public cloud. The more sensitive data get processed on private cloud platform and the open to internet data get processed on public cloud. This improves the confidentiality and reduces the threat of losing the data [19].

Zhangjun in his paper proposes the scheduling policy which fits the market based approach. This paper compares three algorithm Ant colony optimization, Genetic and Particle Swarm algorithm. The paper compares the three algorithms and suggested which is more adaptive for market oriented approach [20].

3. Scheduling Architecture

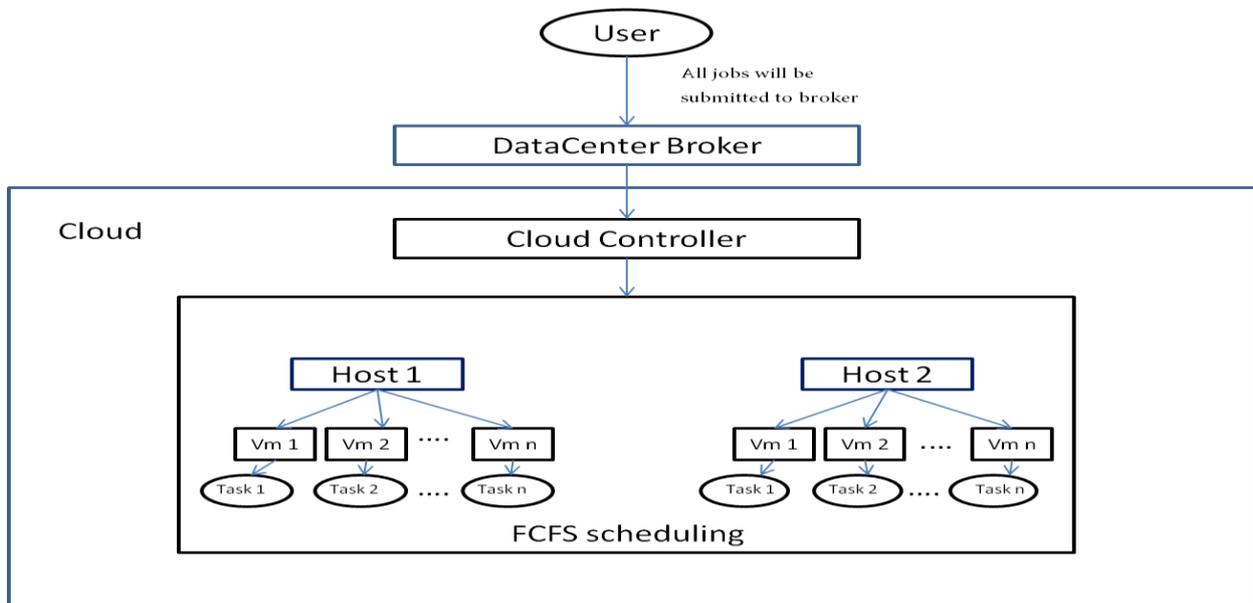


Fig. 1: FCFS scheduling architecture in cloudsim

The cloudsim works in the following way,

At first, user submits all the tasks as cloudlets to datacenter broker. The broker communicates with datacenter through cloud controller who pushes the tasks to datacenter. The datacenter has many hosts and each hosts will be virtualized as different VMs who have their unique specific id. The tasks will be allocated those VMs by following certain scheduling policy. The default scheduling algorithm used by cloudsim is First Come First Serve (FCFS) depicted in Figure 1.

The scheduling algorithm executes by taking the cloudlet length for static tasks where, cloudlet length = cloudlet MIPS * Processing Elements it takes to execute in Datacenter.

This default scheduling algorithm actually takes longer processing/ execution time for the tasks submitted to it.

4. Proposed Algorithm

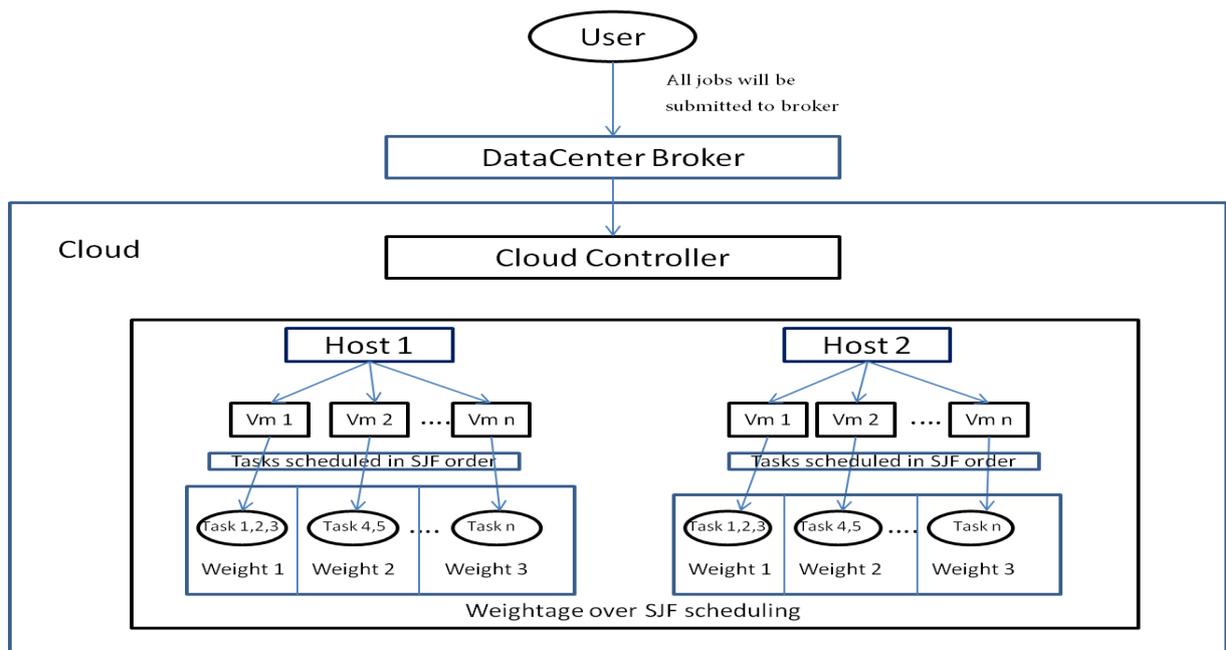


Fig. 2: Weighted SJF scheduling architecture

To reduce the execution time, it is possible to go for Shortest Job First (SJF) scheduling algorithm. But SJF algorithm has a problem making the longest job to starve for resources for long time. The proposed algorithm, additionally gives weightage to the shortest job scheduling algorithm (Figure 2.).

STEP 1: List all cloudlet length and sort in the order of shortest first.

STEP 2: Create 3 categories of weightages 1,2,3 (1-high, 2-medium, 3-low)

STEP 3: Categorise all the cloudlet tasks under these 3 weightages as a queue.

STEP 4: Execute 3 tasks from weightage 1, 2 from weightage 2, 1 from weightage 3.

This reduces starving of resources for long time for the longest job/task. Each job submitted by the user will be executed within short time no matter whether it is a longest job and also utilizes the resources effectively since it also implements SJF algorithm.

5. IOT Component Study

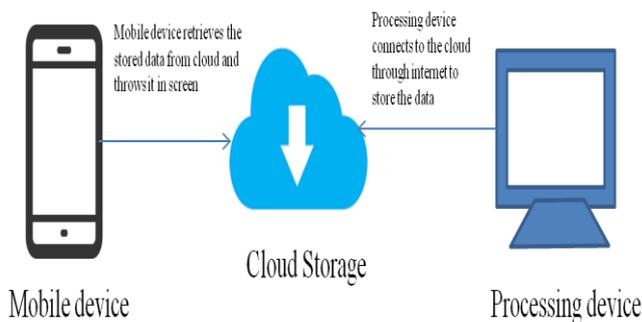


Fig. 3: IOT component used for simulation

Generally, any device which connects to the internet for processing or storage or server can be called as Internet of things (IOT). For an example, let us take mobile device. When the

mobile device connects to the internet for storing or processing the data, we call mobile device as an IOT. In the recent days, many cloud vendors has brought many IOT services as their part. The most popular example is Amazon Web Services (AWS) IOT.

There are also services provided by Amazon Web Services to develop a mobile application. They provide Software Development Kit (SDK) which can be used for developing mobile applications.

In this simulation part, the Amazon Web Services is taken as a cloud vendor. This project uses Amazon free tier services. The components used for simulation are mobile device, AWS Simple Database (SDB) service and the processing element with core processor which contains cloudsim installed as in Figure 3.

Mobile device: The scheduled process simulation results will be shown in mobile device. The tasks which got scheduled through cloudsim will be updated lively in visualization part of the mobile device.

AWS Simple Database (SDB): SDB is a storage service provided by AWS which can store the data. Simple DB simply acts as an indexing mechanism for a data. It has no schema. It simply queries the data based on the index value. It can also integrate with relational Database systems.

Processing device: This device has a cloudsim installed. It has processing elements to process the data. The cloudsim is just a package which can be imported as a jar file in java application. The scheduling algorithm can be coded in cloudsim. The dataset can be feed here for processing.

6. Algorithm Visualization through IOT

Both the mobile device and processing device are made to be connected with the AWS cloud Simple Database (DB1 and DB2). The processing element fetches the static data set from the DB1. The cloudsim in the processing element executes the weightage based scheduling algorithm and produces the scheduled data set result. The resulting dataset will be put into DB2 through java connection with SDB. The mobile device also connects with SDB through the web service. It fetches the data from the Amazon SDB through REST API calls via indexing and querying.

The simulated data will be fetched and projected visually in mobile device as in Figure 4. The scheduled data will be streamed lively and got updated visually through mobile device. The

comparison can be made for this scheduled data produced by weightage based scheduling algorithm and the existing FCFS algorithm.

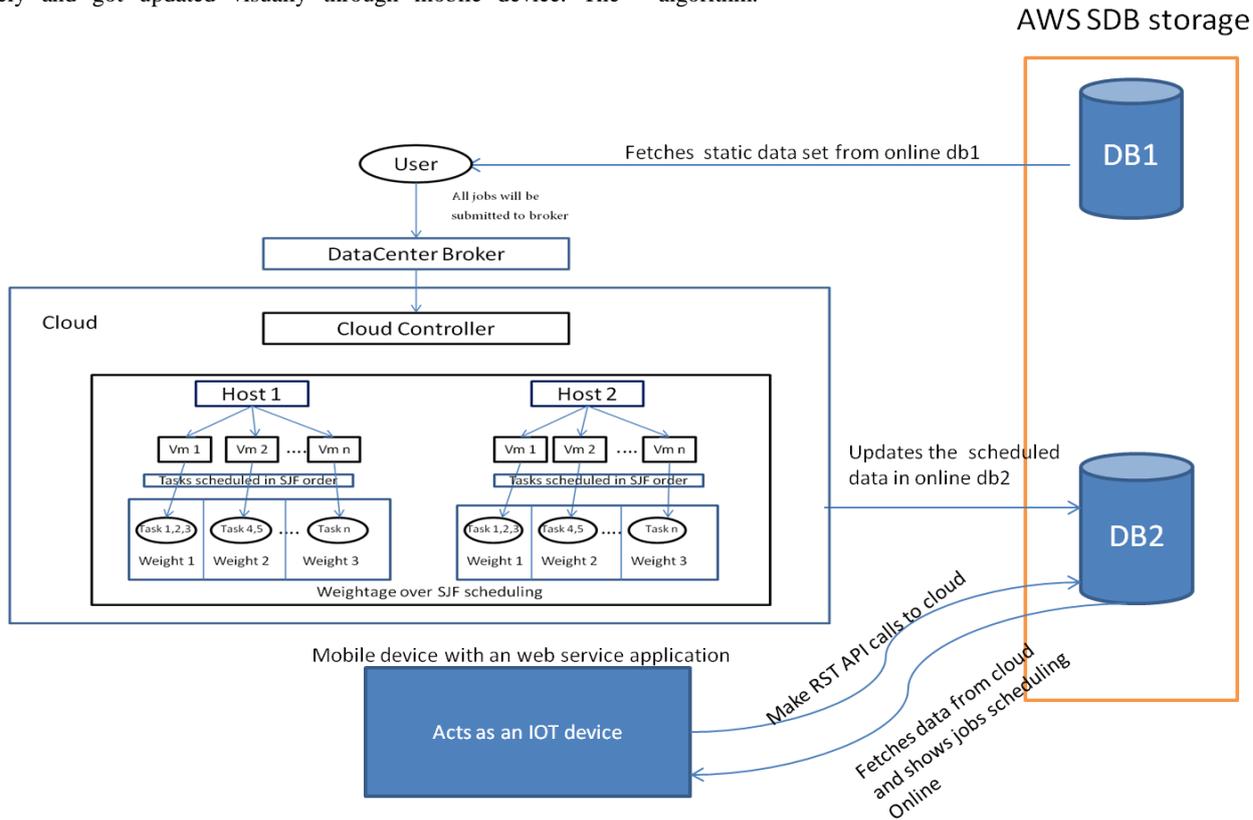


Fig. 4: Simple Database (SDB) connection for the project structure

7. Simulation Results

The simulation here is carried out for static tasks whose processing time is pre-defined. In case of static tasks, cloudlet length is known already and based on that the scheduling algorithm is carried out. The comparison is made for FCFS algorithm which is existing and the proposed Weightage over shortest static job first scheduling algorithm.

The FCFS algorithm doesn't consider the cloudlet length. It executes the task (cloudlet) which arrives first and then move to the next task. But the proposed weightage over SJF algorithm initially sort all the cloudlets submitted to the datacenter broker. Then the sorted cloudlets will be allocated under three weightage queues named weightage 1, weightage 2 and weightage 3. Weightage 1 indicates the cloudlets with short length and the length increases for other weightage queues (weightage queue 2, weightage queue 3). The threshold value for weightage queue 1 for this simulation is set as 0-2000 MIPS and 2001-8000 MIPS for weightage queue 2 and greater than 8000 MIPS for weightage queue 3. In this simulation, 15 cloudlets are submitted to datacenter broker. Each cloudlet will be allocated under any of the 3 weightage queues depending upon the cloudlet length. Then scheduling algorithm runs by executing 3 cloudlets from weightage queue 1 and 2 cloudlets from weightage queue 2 and 1 cloudlet from weightage queue 3. The completion time for each cloudlet is then taken. The simulation results are compared by taking the end time of the cloudlets for both algorithm and are shown as graph in Figure 5.

The comparison result shows that the proposed weightage based scheduling algorithm takes less time to complete the tasks than the existing FCFS algorithm.

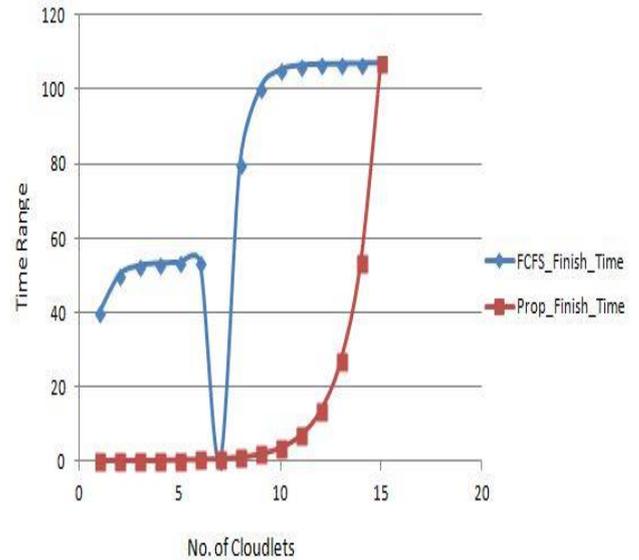


Fig. 5: FCFS Vs Weighted SJF scheduling

8. Conclusion

This paper best analyses the effective scheduling algorithm. Our proposed algorithm gives better output in terms of execution time and also gives priority even for the largest jobs submitted to the datacenter broker. Every task is allocated with resources without making them starve for long time.

References

- [1] Abirami S.P and Shalini Ramanathan, "Linear Scheduling Strategy for Resource Allocation in Cloud Environment", International Journal on Cloud Computing: Services and Architecture(IJCCSA), Vol.2, No.1, February 2012.
- [2] Amdani, S. Y., and S. R. Jadhao. "Novel hybrid cost-priority based scheduling in cloud environment." Recent Advances and Innovations in Engineering (ICRAIE), 2016 International Conference on. IEEE, 2016.
- [3] Bhawna Taneja, "An empirical study of most fit max-min and priority task scheduling algorithms in cloud computing", Computing Communication & Automation (ICCCA) 2015 International Conference on, pp. 664-667, 2015
- [4] Ding, Ding, Xiaocong Fan, and Siwei Luo. "User-oriented cloud resource scheduling with feedback integration." The Journal of Supercomputing 72.8 (2016): 3114-3135.
- [5] Fang Y., Wang F., Ge J. (2010) A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing. In: Wang F.L., Gong Z., Luo X., Lei J. (eds) Web Information Systems and Mining. WISM 2010. Lecture Notes in Computer Science, vol 6318. Springer, Berlin, Heidelberg
- [6] Filip, Ion-Dorinel, et al. "Microservices Scheduling Model over Heterogeneous Cloud-Edge Environments as Support for IoT Applications." IEEE Internet of Things Journal (2018).
- [7] Khalil, Khaled M., et al. "Multiply-Sectioned Bayesian Network for Multi-Agent Learning Based Meta Resources Scheduling in CloudSim" Intelligent Computing and Information Systems (ICICIS), IEEE ,2017"
- [8] Li, Zhongjin, et al. "Cost and energy aware scheduling algorithm for scientific workflows with deadline constraint in clouds." IEEE Transactions on Services Computing (2017).
- [9] Madhumathi, Ramasamy, et al. "Clustering Based User Preference Resource Scheduling in Cloud Computing." International Conference on Smart Trends for Information Technology and Computer Communications. Springer, Singapore, 2016.
- [10] Rawat, Pradeep Singh, et al. "Power consumption analysis across heterogeneous data center using CloudSim." Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on. IEEE, 2016.
- [11] Salehi, M.A. and Buyya, R. "Adapting market-oriented scheduling policies for cloud computing", Proceedings of the 10th Int'l Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2010), Busan, Korea,pp. 351–362 (2010).
- [12] S. Selvarani, G.S. Sadhasivam, "Improved cost-based algorithm for task scheduling in Cloud computing", Computational Intelligence and Computing Research (ICCIC), pp.1-5, 2010.
- [13] Sharma, Murli Manohar, and Anju Bala. "Survey paper on workflow scheduling algorithms used in cloud computing." International Journal of Information & Computation Technology 4 (2014): 997-1002.
- [14] Sidhu, Harmanbir Singh. "Comparative analysis of scheduling algorithms of Cloudsim in cloud computing", International Journal of Computer Applications (0975 – 8887) Volume 97–No.16, July 2014.
- [15] Simão, José, and Luís Veiga. "Partial utility-driven scheduling for flexible SLA and pricing arbitration in clouds." IEEE transactions on Cloud Computing 4.4 (2016): 467-480.
- [16] The NIST definition of cloud computing, NIST special publication 800-145.
- [17] Tsai, Chun-Wei, et al. "A hyper-heuristic scheduling algorithm for cloud." IEEE Transactions on Cloud Computing 2.2 (2014): 236-250.
- [18] Vijindra and Sudhir Shenai. A, "Survey of Scheduling Issues in Cloud Computing", 2012, Elsevier Ltd.
- [19] Wen, Zhenyu, et al. "Dynamically partitioning workflow over federated clouds for optimising the monetary cost and handling run-time failures." IEEE Transactions on Cloud Computing (2017).
- [20] Zhangjun Wu¹, 2, Xiao Liu², Zhiwei Ni¹, Dong Yuan², Yun Yang." A Market-Oriented Hierarchical Scheduling Strategy in Cloud Workflow Systems" in JSC2010.
- [21] S.V.Manikanthan and D.Sugandhi "Interference Alignment Techniques For Mimo Multicell Based On Relay Interference Broadcast Channel " International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume- 7 ,Issue 1 –MARCH 2014.
- [22] T. Padmapriya, V.Saminadan, "Performance Improvement in long term Evolution-advanced network using multiple input multiple output technique", Journal of Advanced Research in Dynamical and Control Systems, Vol. 9, Sp-6, pp: 990-1010, 2017.