
Analytic Hierarchy Process for Resource Allocation in Cloud Environment

C. Revathy^{1,*} and G. Sekar²

¹*P.G. Student, Department of CSE, Adhiparasakthi Engineering College, Melmaruvathur-603319, Tamilnadu, India*

²*Assistant Professor, Department of CSE, Adhiparasakthi Engineering College, Melmaruvathur-603319, Tamilnadu, India*

**Corresponding Author*

Received 24 February 2018; Accepted 15 March 2018;
Publication 30 March 2018

Abstract

Resource management is the primary issue as the demand grows for provisioning resources and computation in cloud systems. When the resource management fails to meet the QoS parameter (response time) requirements, there occurs the due-date assignment problem leading to “resource allocation problem”. This can be optimized by various scheduling algorithms. In this paper analytic hierarchy process is discussed for resource sharing. The results demonstrates that this approach reduces the execution time of algorithm and effectively handles vertical elasticity by adding resources in cloud server. Moreover, the experimental results shows that proposed method can effectively allocates the resources than any other algorithms. Finally, this algorithm is also focused on how efficiency is effectively handled in cloud computing environment while allocating resources since efficiency based resource allocation scheme has been taken into account.

Keywords: Analytic hierarchy process, Resource sharing, Cloud Computing, QoS, SLA.

Journal of Cyber Security, Vol. 7_1, 25–38.

doi: 10.13052/jcsm2245-1439.713

This is an Open Access publication. © 2018 the Author(s). All rights reserved.

1 Introduction

The computation of any distributed system, gave rise to cloud computing that acts as a computational paradigm to meet the requirements of the end users by providing high QoS(Quality of Service) as per the SLA parameters being signed between the customer and the service provider. The three major service layers in cloud are Application layer, Platform layer and Infrastructure layer.

The user data are stored in the Data centres. Users requesting for jobs to the cloud service provider with particular SLA parameters to insist their requirements. The crucial job of the cloud management is, scheduling. Scheduling of user's jobs to the resources that are present in the Infrastructure as a Service layer in the form of VMs (Virtual Machines) are considered as the pre-eminent process to be handled by the service providers in cloud.

Scheduling being an important criteria for Resource Allocation in cloud, various scheduling algorithms are been proposed and are been implemented. The objective of the project is to minimum number of virtual machine is allocated to complete the user tasks within the deadline. To minimize the cost and to improve Customer Satisfaction Level (CSL). Customer driven SLA-based resource provisioning algorithm to minimize cost by maximizing resource and to improve (CSL). Less number of should be selected, to avoid more processing cost for migration process. Migration of Virtual machine tasks to reduce the user provisioning cost, by finishing the user's tasks in their deadline. First, a user sends a request for resources to a provider. When the provider receives the request, it looks for resources to satisfy the request and assigns the resources to the requesting user, typically as a form of virtual machines (VMs). SaaS providers are able to manage the dynamic change of customers, mapping customer requests and handling heterogeneity of Virtual Machines. Enhanced the customer satisfaction level and to improve the service quality the relationship between customer and provider. Quality of Service parameters such as response time, and infrastructure level parameters such as service initiation time. Minimize the SaaS provider's cost and the number of SLA violations in a dynamic resource sharing Cloud environment.

The rest of the article is organized as follows. Section 2 discussed various methods of resource sharing schemes. System model are described in Section 3. Proposed algorithms are discussed in Section 4. The results and performances are compared in Section 5. Finally paper was concluded in Section 6.

2 Related Works

In [1], Service Level Agreement (SLA) is a contractual agreement that has been established between the members of Cloud Service Provider (CSP) that consists of the service provider and its recipient for a specific application of services based on its allocating and sharing Resources as a Service (RaaS). In this context, there are many parties who are very concerning about it, but unfortunately there is lack of a common mapping of SLA information and RaaS of best practice as a guideline that can be used by CSP in making decision for the future purposes in engaging with services in the cloud computing environment. Therefore, the main goal of this paper is to integrate information of SLA and RaaS in order to ensure everybody who are providing and receiving the services in the cloud particularly will be satisfied and getting the best maximum Return Of the Investment (ROI) in allocating and sharing resources among the providers and recipients in cloud computing environment.

The problem of SLA-based joint optimization of application environment assignment, request dispatching from the clients to the servers, as well as resource allocation in a data center comprised of heterogeneous servers are discussed in [2]. The objective is to maximize the total profit, which is the total price gained from serving the clients subtracted by the operation cost of the data center. The total price depends on the average service request response time for each client as defined in their utility functions, while the operating cost is related to the total energy consumption. Experimental results demonstrate that the proposed near optimal joint application environment assignment and resource allocation algorithm outperforms baseline algorithms by up to 65.7%.

In [3], an SLA is a formal contract used to guarantee that consumers service quality expectation can be achieved. In utility computing systems, the level of customer satisfaction is crucial, making SLAs significantly important in these environments. Fundamental issue is the management of SLAs, including SLA autonomy management or trade off among multiple Quality of Service (QoS) parameters. Many SLA languages and frameworks have been developed as solutions; however, there is no overall classification for these extensive works. Service Level Agreements (SLAs) manage the relationship among cloud service providers and cloud consumers by defining the terms of the agreement for the participating entities and provide the basic ground for interactions among both the parties. In [4] they proposed a framework to efficiently monitor and analyze the SLA parameters and tried to find out the possibility of occurrence of SLA violations. Also we implemented an adaptive

resource allocation system by utilizing the results of predicted SLA violations. Our adaptive resource allocation system allocates computing resources to cloud applications and tries to reduce the occurrence of SLA violations, by allocating additional resources on the detection of possibility of occurrence of a violation.

In paper [5–7], the cloud computing is service level agreement which is a contract involving service agreement between client and service computer. According to this contract, violation in commitments mentioned in agreement by service provider will be punished; therefore, service providers should have solutions to reduce these conflicts in service level and costs. Regarding importance of this topic, this article deals with approaches conducted in service level agreement and its comparison regarding improved parameters, simulation or implementation environment, workload and application. In [8–17], various performance enhancement techniques are discussed.

3 System Design

Cloud has three major service layers as AaaS, PaaS, and IaaS. Where the Infrastructure as a Service layer, plays a major role in scheduling. The technological concerns for the resource allocation in IaaS are, Resource management, Performance enhancement and Cloud provider's and cloud user's satisfaction. Scheduling is the measurement of time about which activity should be performed at what time. It is a demand of resources to be delivered to the consignment of tasks to the resources accessible for processing, network and storage, where there is an extreme usage of resources.

Scheduling algorithms are generally used to decrease the execution time. Scheduling tackles the problem of which resources needed to be allocated to the received task or cloudlet. Efficient scheduling algorithms should consider, the total execution time of the available resources, quick recovery and fault tolerance.

Resource allocation in cloud can be handled based on six criteria out of which in this paper, energy aware resource scheduling is taken into consideration. The six criteria are listed below and the parameters included in each criteria are shown in the Figure 1.

In this paper efficiency aware resource scheduling has been taken into account for finding out whether the genetic algorithm can be considered as the most efficient algorithm out of many algorithms.

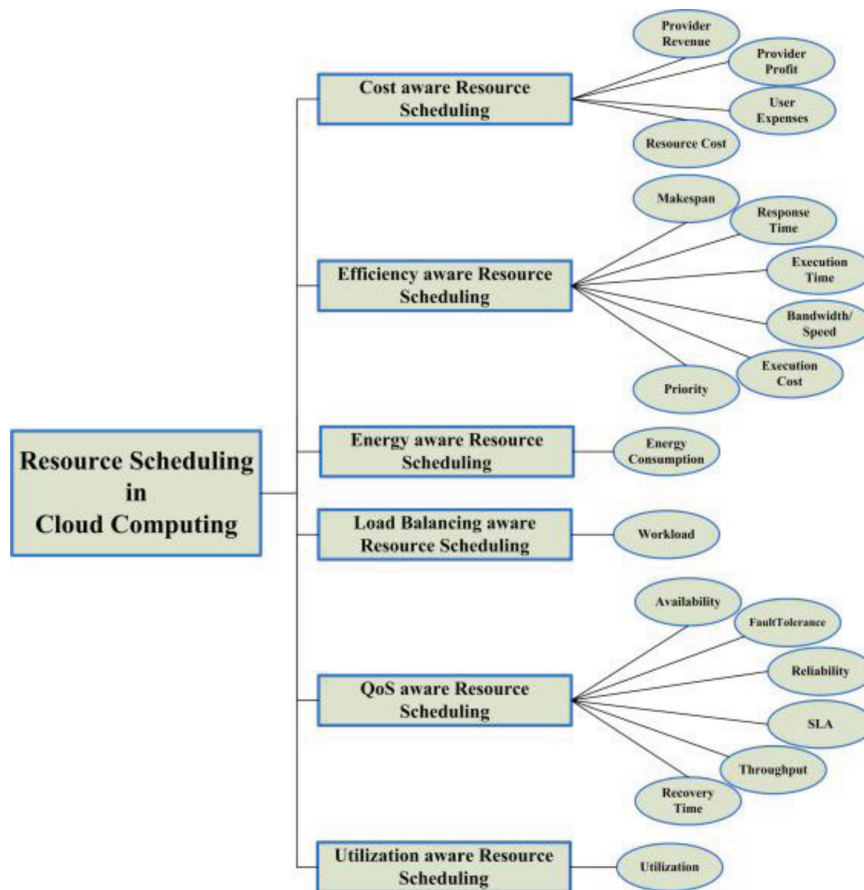


Figure 1 Resource Scheduling in Cloud based on various contexts.

3.1 Efficiency Aware Resource Scheduling

Efficiency aware resource scheduling expresses the amount of resources consumed for processing, depending upon the targeted resources to enhance the efficiency. A scheduling algorithm is said as an effective resource scheduling algorithm when helps to improve and enhance the following factors or parameters: Response time, Execution time, Makespan, Bandwidth/speed and Priority.

The collection of service performance indicates the degree of satisfaction of cloud user for the IaaS resources or services. The desire of cloud users is to acquire a service that should provide the results for the above mentioned

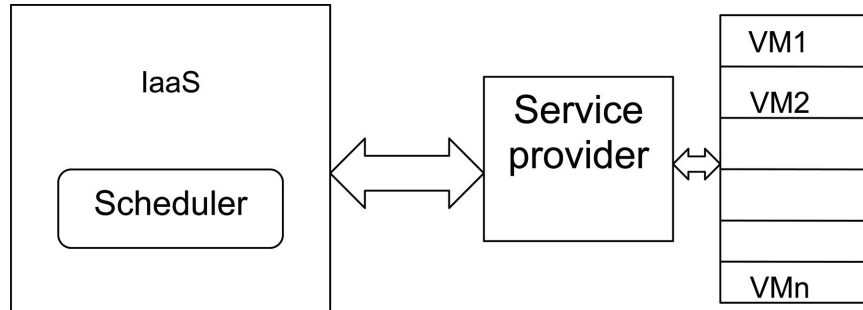


Figure 2 IaaS in CLOUD.

parameters as requested by them. The values for expected response time, execution time and other parameters that are considered for efficiency aware resource scheduling are received by the service provider from a user through SLA (Service Level Agreement). The efficiency depends upon, to what extent the service provider uses the scheduler for satisfying the specified values, at the same time meeting the requirements of the user. Thus the goal is to improve the QoS (Quality of Service). Therefore, improvement of the performance and the quality of services are the most core parameters considered in the environment of cloud computing. Regarding to this fact, the level of efficiency depends on the QoS aware scheduling algorithm and parameters created for the resource and task scheduling in the cloud. The main advantage is that effective resource scheduling can i) minimize the task finishing time and ii) increase the utilization of computing resources in cloud computing. Figures 2 and 3 describes IaaS architecture and system architecture in cloud.

3.2 Definition of Parameters on the Basis of Scheduling in Cloud

- **Response Time:**
It refers to the time, when a cloudlet or task responses to a specific input and start processing. It can be calculated by the sum of waiting time and submission time. It also directly effects on the waiting time of the cloudlet or tasks.
- **Execution Time:**
It determines the time that is consumed by the implementation of cloudlet or task. Minimum execution time is required for cloud provider and user, to enhance the efficiency. It also effects the overall performance.

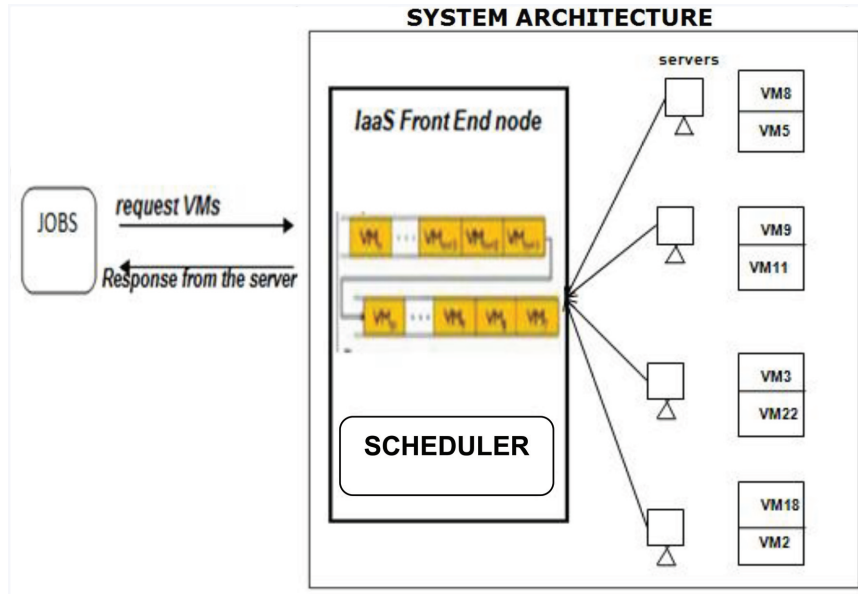


Figure 3 System Architecture.

- **Makespan:**
It determines the maximum completion time of cloudlet or task, when the resources are allocated to the users. So it is necessary to reduce the makespan of specific cloudlet otherwise the request will not be fulfilled on time.
- **Bandwidth/Speed:**
Bandwidth is also defined as the amount of data that can be transferred or executed in a fixed amount of time. It is usually preceded in bits or bytes per second (bps).
- **Priority:**
It refers to a cloudlet or task that is regarded or treated more important than the others.

4 AHP - Modified Approach

The modified AHP method is used to calculate a priority value, called weight, for each task based on its attributes and the user requirements. From the Figure 4, we can see the various steps involved in deriving the priority value or weight used for sorting the various tasks. To make a decision in an organized

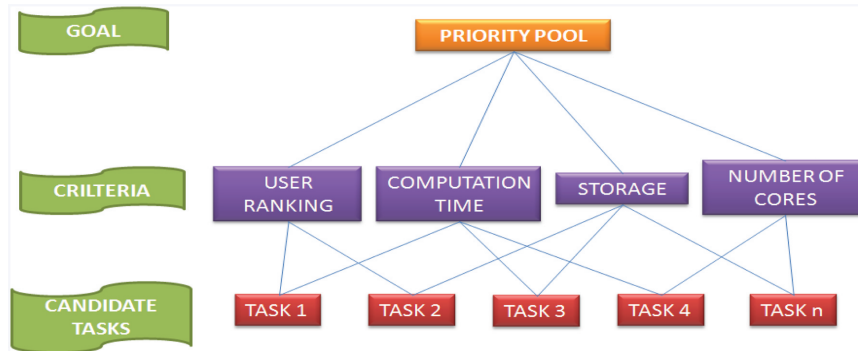


Figure 4 Hierarchy of Selected Criteria.

way and to generate the pairwise matrices, one should have relevant and sufficient knowledge about the application and the cloud environment setup.

- Select the Criteria:

Define the various criteria to be considered by taking into account the user requirements and the capabilities of the available data centers. The Designers must choose the criteria that are important in their viewpoint. In this example we have taken the application requirements (completion time, storage size, computational capacity of the data center hosts) and user requirement (user's task ranking) as our judgement parameters and it is shown in Figure 4.

4.1 AHP Simulation

In this simulation we have imported various classes available in the CloudSim library to create our custom java scheduler class – AhpScheduler, where the actual computational steps for calculating the global priorities is carried out. Apart from this, additional attributes (CloudletPriority and CloudletID) are included in the pre-existing Cloudlet java class. The length of instructions is used as a factor for determining the completion time. The storage space required is determined by the size of the input/output file. The number of processing elements required to complete determines the capacity of computing resource required for the tasks execution. The user's ranking value is also added to shows the importance of the task, based on user's need.

The simulation is run for various scenarios and the results are recorded. Similarly, the same set of scenarios is also simulated for the default FCFS scheduling class present in the CloudSim library files. Using these results we

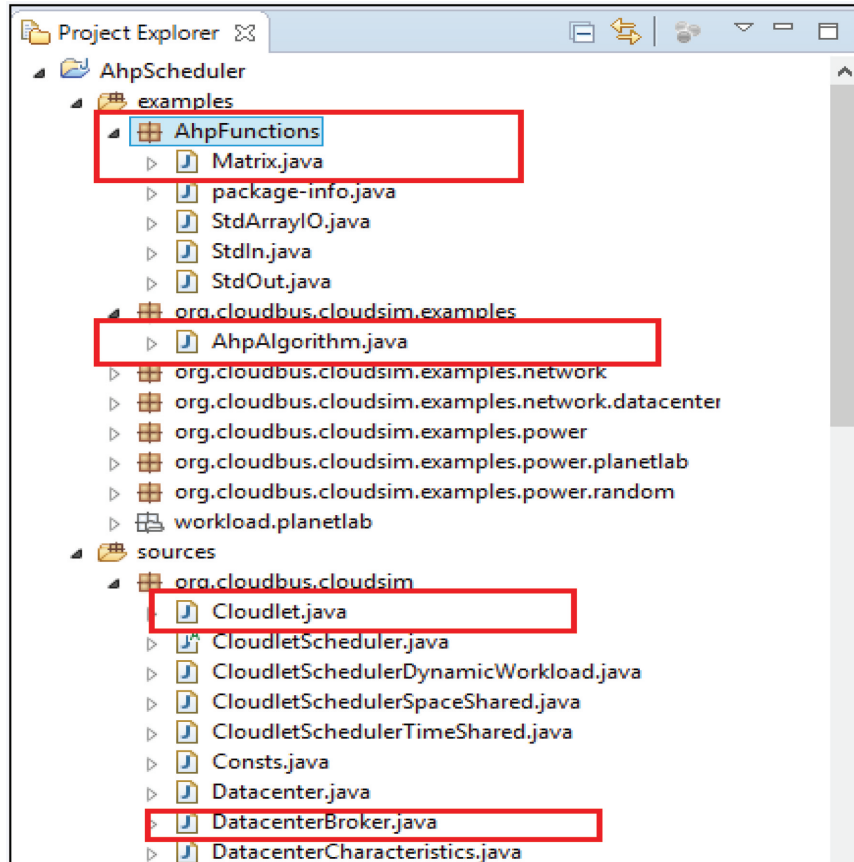


Figure 5 CloudSim Project Setup.

can assess the performance of both the scheduling algorithms. It is illustrated in Figure 5.

5 Results and Comparisons

We have submitted cloudlets ranging from 1000 to 5000 and noted down the total time taken by them to execute with the virtual machine implementing the policies – time shared and space shared, one by one. The outputs are analyzed and the results are visualized using graph. Comparing the total execution time from Table 1 (Space shared) and Table 2 (Time Shared) for Task sets

Table 1 Total Execution Time recorded for Space shared VM allocation policy

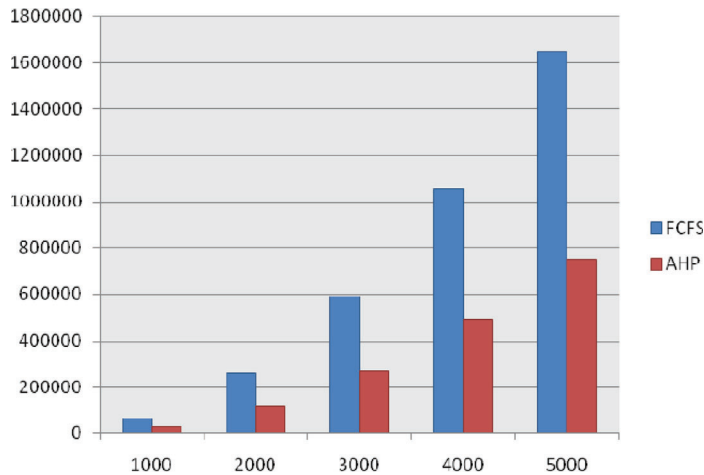
Number of Cloudlets	Total Execution Time (Space Shared)	
	FCFS	AHP
1000	942.421	612.2041
2000	1885.061333	1227.0343
3000	2825.577	1840.763667
4000	3765.875	2452.674633
5000	4709.8505	3083.59

Table 2 Total Execution Time recorded for Time shared VM allocation policy

Number of Cloudlets	Total Execution Time (Time Shared)	
	FCFS	AHP
1000	66044.71467	29899.03604
2000	263895.8304	119690.8755
3000	593432.3197	271685.3025
4000	1055056.647	488940.0065
5000	1648291.749	751045.5264

[1000, 2000, 3000, 4000, and 5000] we can deduce that our modified AHP approach for task scheduling outperforms the default FCFS scheduling.

The line graphs are depicted in Figures 6 and 7, where the number of cloudlets (X axis) is plotted against the total execution time (Y axis) for AHP and FCFS schedulers, using space and time shared VM allocations. The scaling factor for the horizontal axis is taken to be 1000.

**Figure 6** Comparison of Total Execution Time for AHP and FCFS with Time shared VM allocation.

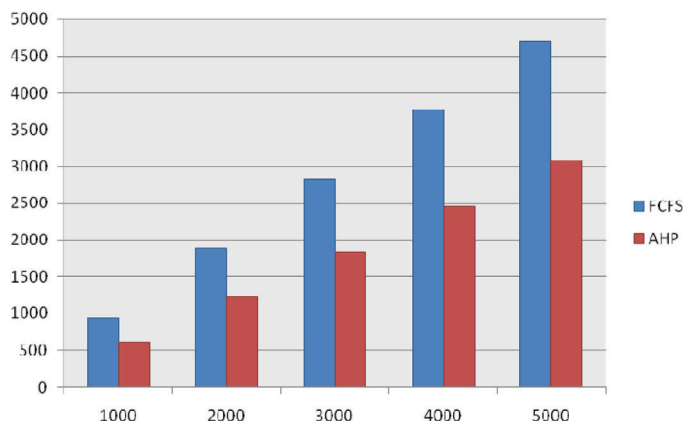


Figure 7 Comparison of Total Execution Time for AHP and FCFS with Space shared VM allocation.

6 Conclusion

Today, most of the organizations are shifting towards cloud computing environment for their different services requirements in order to find an alternative solution for managing on-demand requirement of users. The organizations use rental resources instead of buying additional resources. In this paper, AHP approach is proposed for improving resource allocation in cloud computing environment. It is believed that this approach reduces the execution time of algorithm and effectively handles vertical elasticity by adding resources in cloud server. The AHP has been evaluated by comparing with existing algorithms namely FCFS (First Come First Serve) and round robin. Moreover, the experimental results will show that whether using genetic algorithm can effectively allocates the resources than any other algorithms. Finally, this algorithm is also focused on how efficiency is effectively handled in cloud computing environment while allocating resources since efficiency based resource allocation scheme has been taken into account. The simulation results demonstrate how far the scheduling algorithms works out for the allocation of resources in cloud. For now the simulation tool has demonstrated how the jobs are been scheduled based on genetic algorithm.

References

- [1] Yeo, C. S., and Buyya, R. (2005). Service level agreement based allocation of cluster resources: Handling penalty to enhance utility. In *IEEE International Conference on Cluster Computing*, 1–10.
- [2] Rana, O. F., Warnier, M., Quillinan, T. B., Brazier, F., and Cojocarasu, D. (2008). Managing Violations in Service Level Agreements, In *Proceeding of the 5th International Workshop on GenCon*, Gran Canaris, Spain, 1–10.
- [3] Wu, L., Garg, S. K., and Buyya, R. (2011). SLA-based resource allocation for software as a service provider (SaaS) in cloud computing environments. In *Proceedings of the 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 195–204.
- [4] Fu, Y., and Vahdat, A. (2010). SLA Based Distributed Resource Allocation for Streaming Hosting Systems, Referenced on 6th Dec. 2010.
- [5] Ardagna, D., Trubian, M., and Zhang, L. (2007). SLA based resource allocation policies in autonomic environments. *J. Parallel and Distributed Computing*, 67(3), 259–270.
- [6] Varalakshmi, P., Priya, K. H., Pradeepa, J., and Perumal, V. (2011). Sla with dual party beneficiality in distributed cloud. In *International Conference on Advances in Computing and Communications*, 471–479. Springer, Berlin, Heidelberg.
- [7] Jung, G., and Sim, K. M. (2011). Agent-based adaptive resource allocation on the cloud computing environment. In *40th International Conference on Parallel Processing Workshops (ICPPW)*, 345–351.
- [8] Karthipan, R., Vishvaksenan, K. S., Kalidoss, R., and Babu, R. S. (2016). Uplink capacity enhancement in IEEE 802.22 using modified duplex approach. *Wireless Personal Communications*, 86(2), 635–656.
- [9] Rajakani, K., and Bhagyaveni, M. A. (2012). Adaptive duplex technique for reduction of turnaround time in IEEE 802.22. In *Global Trends in Computing and Communication Systems*, 503–511. Springer, Berlin, Heidelberg.
- [10] Dhevi, B. L., Vishvaksenan, K. S., and Rajakani, K. (2018). Isolation Enhancement in Dual-Band Microstrip Antenna Array Using Asymmetric Loop Resonator. *IEEE Antennas and Wireless Propagation Letters*, 17(2), 238–241.
- [11] Nelson, I., Annadurai, C., Kalidoss, R., and Partibane, B. (2017). Mitigation of co-channel interferences in cognitive multi-carrier code division multiple access system by singular value decomposition techniques. *Cluster Computing*, 1–7.

- [12] Partibane, B., Alsath, G. N. M., and Rajakani, K. (2017). Design of a bandwidth enhanced hybrid slot loop antenna for GSM/UWB standards. *Circuit World*, 43(3), 105–110.
- [13] Vishvakshnan, K. S., Mithra, K., Kalidoss, R., and Karthipan, R. (2016). Experimental Study on Elliott Wave Theory for Handoff Prediction. *Fluctuation and Noise letters*, 15(4), 1650025.
- [14] Vishvakshnan, K. S., Kalaiarasan, R., Kalidoss, R., and Karthipan, R. (2018). Real time experimental study and analysis of Elliott wave theory in signal strength prediction. In *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, 88(1), 107–119.
- [15] Karthipan, R., Vishvakshnan, K. S., Kalidoss, R., and Krishan, A. (2016). Performance of turbo coded triply-Polarized MIMO–CDMA system for downlink communication. *Computers & Electrical Engineering*, 56, 182–192.
- [16] Kalidoss, R., Bhagyaveni, M. A., and Ahmed, A. H. (2015). A Location Based Duplex Scheme for Cross Time Slot Interference Reduction in (IEEE 802.22) Cognitive Radio Based Wireless Regional Area Networks (WRAN). *Arab Gulf Journal of Scientific Research*, 33(1).
- [17] Kalidoss, R., Bhagyaveni, M. A., and Vishvakshnan, K. S. (2014). A location-based duplex scheme for cost effective rural broadband connectivity using IEEE 802.22 cognitive radio based wireless regional area networks. *Fluctuation and Noise Letters*, 13(4), 1450028.

Biographies



C. Revathy completed bachelor degree (B.E., 2004) in Computer Science and Engineering from Madurai Kamaraj University. Prior to her joining the Masters programme, she has worked at Cognizant Technology Solutions, Chennai. Currently she is pursuing Master of Engineering in Department

of CSE, Adhiparasakthi Engineering College, Melmaruvathur. Her research interests cover Cloud computing, high performance computing, Distributed computing, Data Mining, Bioinformatics, Machine Learning and Pattern Discovery.



G. Sekar received his B.Tech. degree in IT from Anna University, Chennai, in 2005, and the M.E degree in CSE from Anna University of Technology, Tirunelveli, in 2010. He currently holds an academic post as Assistant Professor in the department of CSE, Adhiparasakthi Engineering College, Melmaruvathur. His research interests include Data structures, Computer organization and architecture, Computer Networks, Operating systems, Mobile and Pervasive Computing.