SCITECH
RESEARCH ORGANISATION|

Volume 3, Issue 1
March 30, 2015|

Journal of Information Sciences and Computing Technologies
www.scitecresearch.com

Scheduling Algorithms for Cloud: A Survey and Analysis.

Prakash Kumar[1], Krishna Gopal[2], J P Gupta[3]
[1&2]Department of Computer Science Engineering & IT, JIIT, Noida, India
[3]Hydrocarbons Education and Research Society, New Delhi.

Abstract

Cloud Computing is a fast growing computing paradigm as it provides vast benefits to the users. Scheduling becomes one of the key aspects due to the pay-as-you-go nature of the Cloud. The factors affecting the technique of scheduling applied change with change in scenarios. For scheduling in hybrid clouds, the data transfer speed has to be taken into consideration whereas for mobile environments scheduling becomes dependent on context change. Moreover scheduling can be improvised on many fronts such as energy efficiency, cost minimization, Maximization of resource utilization, etc. This paper surveys scheduling techniques in various Cloud Computing scenarios and sites the most efficient scheduling technique available for a particular set of user needs by comparing various techniques and the problems they address.

Keywords: Cloud Computing; scheduling algorithms; resource allocations; QoS; Workload testing.

1. Introduction and motivation

Cloud Computing is spreading out at a very high speed among the IT companies due to the cost saved up in infrastructure and reduction in the cost of IT management. Cloud Computing is a huge field and it has various aspects on which its efficiency is dependent, a very important one being scheduling. Scheduling is a major factor needing attention in the field of cloud computing. Amount of energy consumed, cost incurred to provide services over the cloud, amount of execution time, these are major causes of concern and improvising the scheduling of tasks helps in minimizing these. A lot of research has taken place in this field .The stipulation to maintain the QoS delivered to the user makes the job of scheduling tasks on the resources even more complex. Techniques such as VM resizing introduced by the EnaCloud algorithm and the neural predictor put forward by the green predict scheduling algorithm help in minimizing the energy consumption because it is a major cause of concern in today's time. Also users expect the makespan of tasks to be as small as possible. Algorithms such as Hyper-heuristic scheduling algorithm aim at reducing the execution time of tasks. Moreover the parameters to be taken into consideration change with a change in the type of applications running on the cloud, the type of resources provided by the cloud and the presence of private infrastructure. This paper surveys scheduling techniques which help in efficient management of resources in different scenarios and concludes by suggesting which scheduling technique is it for a particular scenario on the basis of user need and the characteristics of the Cloud available.

2. Related Work

Liang Luo et al.[1] put forward a strategy to save energy in case of tasks which don't use the whole of the hardware allocated to them. Allocation of unnecessary components is prevented by this strategy. The components can be categorized as CPU, storage, memory and network. The various tasks come under IO intensive, CPU intensive and network intensive tasks. Tests showed that IO intensive tasks consume very less of CPU and majorly work using memory and storage whereas IO tasks over a network use a small percentage of CPU and network apart from memory and storage. Thus according to the specific needs of the task types component policies are generated. Allocation of components is done according to the task type using predefined policy.

Bo Li et al.[2] proposed a strategy to save energy wasted due to servers sitting idle during the placement of applications with constantly varying requirements. The key to the algorithm is that a task is enveloped by a VM to enable transferring of the task to another resource while the task is in progress. This helps in reducing the number of active servers. While insertion the task at hand replaces a smaller task if it can and the smaller one is now inserted in the same manner. The principle behind this strategy is that smaller tasks can be easily accommodated into already active servers as compared to bigger tasks. When a task is to be deleted it releases the resources and tasks on that server go through insertion process. If they can be inserted in the already active servers that server's resources are released altogether and the server is closed. If a task undergoes resizing it first releases its current resources and then undergoes insertion. To avoid very frequent migrations in turn leading to wastage of energy an energy aware heuristic is proposed by the paper. An over provision approach is also put forward to deal with the frequently changing needs of the task and reduce unnecessary migrations.

Truong Vinh Truong Duy et al [4] presented a Green scheduling algorithm based on the decision made by a neural network predictor put forward by the paper. It was observed that powering off the servers while they were not in use saved a lot more energy as compared to lowering the voltage. The problem encountered was to assess the future demand and power off the extra servers as per the assessment because in case of a wrong prediction the drop rate of user requests increases and service level agreement is not assured. The neural network predictor put forward by this paper gives a solution to this problem. It assesses the need of number of servers in the future based on the previous records. The predictor works in four modes namely optimal mode, conventional mode, predictor mode and predictor mode with additional 20% servers. Tests proved that predictor mode with additional 20% servers proved to be the best as it helped to save the maximum amount of energy at minimum drop rate.

Marco A.S.Netto et al.[3] proposed that devices running applications such as voice assistants and shopping assistants should have components which can find out the user context specific to that application. This would save many resources as, in applications such as these the results become worthless if the user context changes and new results have to be generated for the new user context. Thus computation spent to process the information with respect to users old context is wasted. A concept of window of opportunity was also defined by the paper which is the duration after the user enters a particular context in which a result is valid and outside it the result is useless to the user. It was inferred that for a result to be useful the execution time of the task has to be less than or equal to the window of opportunity thus tasks getting completed outside the window of opportunity are aborted reducing wastage of resources. It was noticed that for less resources, utilization of resources decreased due to cancellation of tasks lying out of the window of opportunity and the utilization kept on increasing with the increase in resources up to a certain threshold after which it became same as in the absence of context.

Yiqiu Fang et al. [5] proposed a scheduling algorithm for computational tasks. The algorithm uses load balancing to increase the resource utilization ,at the same time meeting user's requirements. The algorithm calculates a value which evaluates the load. This value is based upon the execution time of the running tasks, number of virtual machines and number of hosts.

The lower the value the better is the state of the system and after every iteration this value is checked. If it is greater than the maximum acceptable value then the load is balanced by migrating virtual machines to the host having less load and thus striking a balance in the whole system. This algorithm took into consideration the feature of a variable user demand found in Cloud Computing and tests proved that it is successful in increasing the resource utilization.

Tien Van Do et al.[6] compared various scheduling algorithms to find out an efficient solution for the allocation of servers to fulfill users' demand at the same time taking into consideration the energy consumption. The key feature of this algorithm was the study of the relationship between the user demand and servers. Tests revealed that for a small load least first policy works out the best in terms of energy and heat emission without a very large rejection rate of user requests whereas for medium loads priority policy for servers works better wherein the larger the server higher is its priority. Also the order in which the servers are set plays an important role. The servers with maximum amount of resources should be given the highest priority.

Ruben Van den Bossche et al.[7] proposed an algorithm to schedule deadline constrained tasks in hybrid clouds within the minimum cost while maintaining the QoS. Options are available for IT companies to use their pre-existing infrastructure along with public cloud services in the form of a hybrid cloud. Four scheduling policies were put forward by combining the two queue policies EDF and FCFS with two cases-sending the cheapest task to public cloud and sending the unfeasible task to public cloud. They were compared with each other and with two other cases, first being the case where all tasks are scheduled on the public cloud and second a Cost Oriented Scheduling policy introduced by the author in previous work. Tests show that sending cheapest task using Earliest deadline first stood out in terms of performance as in this case less data intensive costs are sent to the public cloud due to deduction of data transfer cost and this also helps in meeting deadlines ad data transfer time is saved.

Jiayin Li et al.[8] proposed two algorithms named Dynamic Cloud List Scheduling(DCLS) and Dynamic Cloud Min-Min Scheduling(DCMMS) in which tasks are categorized in two types namely Advanced Reservation(AR) and Best Effort. The AR tasks are given a higher preference and to run them Best effort tasks are preempted. The two algorithms were proposed for scenarios where workload is heavy and applications tend to have conflicts over resources. These algorithms increased system utilization and decreased energy wastage in such situations. The algorithms accommodates any changes in scheduling which may occur when the task is actually executed and this property is successful in reducing a considerable amount of conflicts in case of heavy traffic. Tests showed that DCMMS gave a better performance as compared to DCLS and the mapping in the algorithm shows a major improvement in energy efficiency.

Saied Abrehami et al.[9] proposed two scheduling algorithms which were a modification of a grid computing resource allocation algorithm named Partial Critical Paths. Three main differences between Cloud Computing and Grid Computing were taken into consideration. They are:

a) In Cloud Computing the users are given the freedom to demand limitless resources.

b) The concept of pay for what you use

c) Use of heterogeneous bandwidths in case of Cloud Computing

Two modified algorithms were proposed, first being IaaS Cloud Critical Partial Paths (ICPCP) which is a one phase algorithm where each partial critical path is directly scheduled. Second algorithm IaaS Cloud Partial Critical Path with Deadline Distribution (ICPCPD2)which is a two phase algorithm in which in the first phase tasks are given sub deadlines depending on the deadline for the application and in second phase each task is allocated a server. Tests revealed that ICPCP proved to be more efficient as compared to ICPCPD2.

Maria A. Rodriguez at al. [10] proposed a met heuristic algorithm which used Particle Swarm Optimization to schedule tasks on virtual machines. The algorithm took into consideration all the key features of scheduling in IaaS such as variability in user demand, heterogeneity, meeting of QoS, and change in virtual machine characteristics. In Cloud Computing the resources on which the tasks are to be mapped are variable and are not known in advance, unlike in Grid Computing. Thus the task scheduling becomes a two-step problem in case of Cloud Computing because prior to allocation of resources to the tasks the resources have to be assessed. This algorithm merges these two steps into one problem and finds an optimal solution wherein both the tasks are handled by the algorithm. Tests have proved that the algorithm is successful and overpowers other algorithms such as IC-PCP.

Chun-Wei Tsai et al.[11] proposed a hyper heuristic algorithm which with the help of two operators namely diversity detection operator and improvement detection operator made a decision as to which low level heuristic has to be applied depending upon the strengths and weaknesses of the heuristics, and also when to change a heuristic. The low-level heuristics from which the algorithm chooses are particle swarm optimization [10], ant colony optimization [12], simulation annealing[14] and genetic algorithm[13]. Tests show that this algorithm out performs general heuristic algorithms and also the computation time of the algorithm is not large as in every iteration only one heuristic algorithm is run unlike in hybrid-heuristic algorithms[5] which combine two or more heuristic algorithms to find the optimal result.

Yue Gao et al.[15] proposed a strategy to decrease the occurrence of soft errors at the same time taking into consideration the energy efficiency of the system. It includes two schedulers static and dynamic. Soft errors are caused due to noise, high energy cosmic particles and hardware fatigue. Soft errors may lead to a corrupted output or a system crash decreasing the user's QoS. To increase the QoS for the users these soft errors have to be hidden from them. This can be done by predicting and preventing these errors using methods to detect errors and make the system fault tolerant. Some such methods are Virtual Machine Replication [16] or Idempotent Task Try [17]. The drawback of these methods is that they cost a lot in terms of energy. This paper put forward an integrated algorithm to make the system fault tolerant and energy efficient at the same time. The paper put forward a strategy wherein the Cloud Service Provider balances between increase in systems fault tolerance and decrease in the energy consumption. Tests proved the approach to be correct as it lead to 50% reduction in failure rate at an overhead of 76% in terms of energy.

Multi-objective workflow analysis framework proposed by Orachun Udomkasemsub et al.[1] puts forward an algorithm which gives an optimal solution in the case when the user requires the task scheduling to meet various objectives at the same time such as cost minimization along with reduction of make span. The algorithm uses Artificial Bee Colony method. The paper on the basis of previous researches states that Artificial Bee Colony method is found out to be better than other algorithms of the same category such as Particle Swarm Optimization and Genetic algorithm. It was also tested and compared with HEFT and HEFT/LOSS and proved to be better than them in terms of meeting multiple objectives set by the user.

3. Analysis of cloud algorithms

Each algorithms address one or two of the parameters on the basis of which scheduling is improved. Some target cost optimization whereas others address shortening the make span whereas some work at minimizing the overall energy consumption. At the same time these algorithms struggle to meet the Service Level Agreement and satisfy the users by maintaining the Quality of Service. Different Cloud Structures demand different scheduling strategies.

The details regarding the various testing parameters and the criterion for testing for the algorithms surveyed are cited in Table 1.

Sl. No.	Algorithm	Target environment	Criterion	Simulation environment	Testing Workload	Load Size	Comparisons and Comments
1.	Context Aware Scheduling	Mobile application requests running on cloud	QoS improvement and reduction of resource wastage	Event Driven Simulator	Requests for applications such as shopping assistants and voice assistants.	Peak load- keeps tipping during work hours	Successful for a normal workload as well but works best for peak days.
2.	Energy Aware Fault Tolerant Framework	Public Cloud	Reduction in soft errors and maintenance of energy efficiency	Runtime Simulation Engine	Deadline Sensitive Workloads.	Large	Better in terms of energy efficiency as compared to Triple Modular Redundant System.[3]
3.	Green Scheduling Algorithm with neural predictor	Public Cloud	Improve Energy Efficiency and Minimize Drop Rate.	CloudSim and GridSim	Generated Workloads same as requests to NASA and ClarkNet web servers.[1]	Copies a normal day's trend i.e. large during working hours and small at the start and end.	Four modes are introduced among which prediction along with additional servers is most successful for all workload tested.
4.	EnaCloud	Public Cloud	Improve Energy Efficiency	iVic and Xen hypervisor	Web server and Data Servers ,Compute-Intensive ,Common applications	NA	10 % more energy efficient as compared to FCFS and 13% to Best Fit
5.	Energy Efficient Optimization Method	Public Cloud	Improve Energy Efficiency	Real System	Tasks which don't use the whole of the hardware	NA	It works efficiently in case of I/O intensive tasks wherein CPU usage is negligible.
6.	Least first	Public Cloud	Energy efficiency keeping QoS in mind	Numerical Study	NA	small	A higher priority is given to servers with a larger capacity.
7.	Priority					medium	
8.	IC-PCP	Public Cloud	Cost Minimization within Deadline		Scientific Workflows	NA	SCS and PSO are almost equally efficient in meeting deadlines but PSO incurs a smaller cost as compared to SCS. Both of them perform better when deadlines are relaxed. ICPCP misses out on deadlines in most cases.
9.	PSO			CloudSim			
10.	SCS						
11.	Hyper-heuristic Scheduling Algorithm	Public Cloud	Reduction of make span	CloudSim and Hadoop	Workflow and hadoop map-task	NA	Better as compared to hybrid heuristic algorithm due to less computation and case specific algorithm.
12.	Load	Public Cloud	Maximize	CloudSim	Computation	NA	Attains maximal utilization

	Balancing Scheduling		resource utilization and meet QoS		al Tasks		of available resources
13.	Earliest Deadline First (EDF)	Hybrid Cloud	Cost Minimization with Deadline Constraints	Java Based Discrete Time Simulator	Batch Type Workloads with Deadlines	Large	EDF overpowers FCFS as it successfully does the task within Deadlines taking into consideration the data transfer speeds and also handles runtime estimation errors successfully.
14.	FCFS					Small	
15.	Dynamic min-min Scheduling	Multiple Clouds with Heterogeneous Resources	Energy efficient resource scheduling among multi-cloud heterogeneous system	Cloud Simulation Environment (developed by the author)	Workloads same as requests in Parallel Workload Archives.[2]	Large	DCMMS handles situations with large workloads resource contention occurs due to greater number of Advanced Reservation jobs better. Its also more efficient in terms of energy.
16.	Dynamic Cloud List Scheduling					Large	

Table 1: Critical Analysis of the existing Cloud Algorithms.

Context Aware job scheduling algorithm works for scheduling in mobile environments. It reduces resources wasted due to computation of unnecessary requests, those which are no longer useful to the user or the ones which can't be completed in the window of opportunity. The algorithm was tested in three conditions – normal, peaky and flat. Normal symbolized a normal working day wherein the maximum number of requests is concentrated in the working hours, peaky is for scenarios where in the working hours number of requests keeps on increasing at a high rate and dropping after sometime to rise yet again. Flat is to set a baseline wherein number of request remains constant throughout the day. This algorithm works best for peak cases. On increasing the amount of resources the utilization of resources increases to a certain extent in the presence of context after which on further increase in resources utilization follows the same trend as in the absence of context. To make the Energy Aware Fault Tolerant Framework maybe applied. It works better as compared to Triple Modulated Redundant [18] System in terms of energy overhead. To save energy is the need of the hour. Thus, many algorithms aim at minimizing the amount of energy consumed while not compromising the quality of service. Enacloud, Green Predict, Energy Optimization Framework all aim at increasing the energy efficiency. Enacloud proposes an algorithm which helps in scheduling the tasks on the minimum number of servers such that number of active servers is minimal. Green Predict helps in predicting the number of servers which will be needed at a particular time depending upon previous data. It has four modes normal, optimal, predict ad predict with additional servers. Predict with an additional 20% servers proved to be the best mode for energy saving. Energy Optimization framework works well in case hardware is not completely utilized by a task and helps in avoiding allocation of unnecessary resources to a task. Also Tien Van do et al[] proved that prioritizing servers with larger capacity is beneficial to save energy in case of medium loads whereas for small load allocating resources from server having least capacity first is more beneficial.

To minimize cost in case of deadline constrained scientific workflows algorithms such as IaaS Cloud Partial Critical Path, Particle Swarm Optimization and SCS are available. PSO and SCS prove to be effective in minimizing cost while finishing the task within deadline whereas IC-PDP is unsuccessful in completing the tasks within deadline in most cases. The working of PSO and SCS improves when deadlines are relaxed. Between SCS and PSO,PSO tends to complete the task at a lower cost although it takes more time as compared to SCS but it successfully finishes the task within deadline unlike IC-PDP.In hybrid clouds minimization of cost for deadline constrained batch workloads in which the decision of sending a task to the private or public cloud is done by sending the cheapest task to the public cloud as this gives less data intensive tasks to the cloud. This in turn helps in easy completion of tasks within the deadline because data intensive tasks are run in the private cloud thus saving data transfer time. Queuing is done using Earliest Deadline First as it handles error in runtime estimation time better. For scheduling in Multiple Clouds with heterogeneous resources in which two kinds of tasks are present- Advanced Reservation and Best Effort, Dynamic Cloud MinMin Scheduling performs better as compared to Dynamic Cloud List Scheduling. It is better in terms of energy efficiency while meeting the QoS. In cases where user needs the algorithm to meet multiple objectives for example reduction of cost along with make span,artificial bee colony method can be applied.

4. Conclusion

Our paper surveys the various scheduling algorithms in different scenarios. Scheduling can be optimized on the basis of many factors such as energy efficiency, cost reduction, make span reduction. Constraints such as maintaining QoS and Service Level Agreement by the Cloud Provider makes scheduling of task a complex job. From the above survey we can conclude that in different scenarios different scheduling algorithms are applicable. For obtaining energy efficiency in mobile environments such as shopping assistants and voice assistants Context Aware Job Scheduling can be applied. Frameworks can be introduced for energy efficient scheduling of resources such as EnaCloud, green predict and energy optimization resource allocation wherein resources are allocated based on the type of task at hand. These frameworks can also be tested in combination. Green Predict foretells the amount of servers to be kept in an active mode such that number of idle servers is reduced whereas Ena Cloud places the tasks such that minimum number of servers is kept open. Thus the neural predictor of Green Predict can be used in collaboration with the scheduling algorithm of EnaCloud. Also energy optimization resource scheduling can be introduced in the above approach preventing the allocation of unnecessary resources to the tasks.

For tasks with deadlines, to schedule tasks in minimum cost Earliest Deadline first can be chosen for batch workloads whereas Particle Swarm Optimization can be chosen for scientific workloads. Make span of the tasks can be reduced using hyper heuristic algorithm which proves to be better than hybrid heuristic algorithm. For a multicloud system Dynamic Cloud Min-Min Scheduling may be applied. Fault Tolerant Scheduling with moderated energy consumption can be achieved using Energy Aware Fault Tolerant Scheduling Framework. Cloud Computing has many parameters and the scheduling to be applied depends upon user requirements. The aim of any scheduling algorithm is to meet used demand with minimum overheads.

References

[1] Liang Luo,Wenjun Wu, Dichen Di,Fei Zhang,Yizhou Yan,Yaokuan Mao*,"A resource scheduling algorithm of cloud computing based on energy efficient optimization methods"*, International conference on Green Computing, pp: 1-6, 2012.

[2] Bo Li, Jianxin Li, Jinpang Huai, Tian yu Wo, Qin Li, Liangh Zhong*,"EnaCloud-An Energy-Saving Application Live Placement Approach for Cloud Computing Environments"*, CLOUD'09. International IEEE Conference Cloud Computing, pp: 17-24. 2009.

[3] Marcos D Assuncao, Marco AS Netto, Fernando Koch, Silvia Bianchi*,"Context-Aware Job Scheduling for Cloud Computing Environments"*, 5th International IEEE Conference on Utility and Cloud Computing(UCC), pp: 255-262, 2012.

[4] Truong Vinh Truong Duy, Sato Y.,Inoguchi Y.*,"Performance evaluation of a Green Scheduling Algorithm for energy savings in Cloud Computing"*, International Symposium Workshops and Ph. D. forum on Parallel and Distributed Processing, pp: 1-8. 2010.

[5] Yiqiu Fang,Fei Wang, Junwei Ge*,"A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing"*, Proceedings of the International Conference on Web information systems and mining, pp: 271-277, 2010.

[6] Tien Van Do, Csaba Rotter*,"Comparison of scheduling schemes for on-demand IaaS requests"*, Journal of Systems and Software, Volume 85: Issue 6, pp: 1400-1408, 2012.

[7] Van den Bossche R.,Vanmechelen K.,Broeckhove J.*,"Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds"*, 3rd IEEE International Conference on Cloud Computing Technology and Science (CloudCom), pp: 320-327, 2011.

[8] Jiayin Li, Meikang Qiu, Zhong Ming, Gang Quan, Xiao Qin, Zonghua Gu*,"Online optimization for scheduling preemptable tasks on IaaS cloud systems"*, Journal of Parallel and Distributed Computing, Vol: 72, Issue 5, pp: 666-677, 2012.

[9] Saeid Abrishami, Mahmoud Naghibzadeh, Dick H.J. Epema*,"Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds",* , Journal of Future Generation Computer Systems Pages: Vol. 22: Issue 1, pp: 158-169, 2013.

[10] Maria A. Rodriguez,Rajkumar Buyya*,"Deadline based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds"*, IEEE Transactions on Cloud Computing, Volume: 2, Issue 2, pp: 222-235, April, 2014.

[11] Chun-Wei Tsai, Wei-Cheng Huang, Meng-Hsiu Chiang, Ming-Chao Chiang, Chu-Sing Yang,*"A Hyper-Heuristic Scheduling Algorithm for Cloud",* IEEE Transactions on Cloud Computing, Vol: 2, Issue 2, pp. 236-250, April, 2014.

[12] Tawfeek M.A.,El-Sisi A.,Keshk A.E., Torkey F.A.,*"Cloud task scheduling based on ant colony optimization"*, 8th International Conference Computer Engineering and Systems, pp: 64-69, 2013.

[13] Yibin Wei,Ling Tian,*"Research on cloud design resources scheduling based on genetic algorithm",* International Conference on Systems and Informatics, pp:2651-2656, 2012.

[14] Pandit D., Chattopadhya S.,Chattopadhyay M.,Chaki N.,*"Resource allocation in cloud using simulated annealing"*,in Applications and Innovations in Mobile Computing, pp: 21-27, 2014.

[15] Yue Gao,Gupta S.K.,Yanzhi Wang,Pedram M.,*"An Energy-Aware Fault Tolerant Scheduling Framework for Soft Error Resilient Cloud Computing Systems",* Conference and Exhibition on Design, Automation and Test in Europe, pp: 1-6, 2014.

[16] http://www.vmware.com/products/fault-tolerance.

[17] J. Li et al., *"Fault tolerance and scaling in e-Science cloud applications: observations from the continuing development of MODISAzure,"* 6th International Conference on e-Science, pp. 246-253, 2010.

[18] Ricardo Paharsingh, Olivia Das, *"An Availability Model of a Virtual TMR System with Applications in Cloud/Cluster Computing,"* 13th IEEE International Symposium on High-Assurance Systems Engineering, pp. 261-268, 2011.

[19] Traces in the Internet Traffic Archive, http://ita.ee.lbl.gov/html/traces.html.

[20] Parallel workloads archive, www.cs.huji.ac.il/labs/parallel/workload/.

[21] Udomkasemsub).,Li Xiaorong, Achalakul T., *"A multiple-objective workflow scheduling framework for cloud data analytics"* IEEE International Joint Conference on Computer Science and Software Engineering, pp. 391-398, 2012.

Authors' information

Prakash Kumar has received his B. Tech. in Electronics and Communication Engineering, and M. Tech in Computer Science and Technology from University of Roorkee (Now Indian Institute of Technology, Roorkee), India. He is currently an Assistant Professor (Senior Grade) in Jaypee Institute of Information Technology,(Deemed University), Noida, India. His area of interest is in Computer Networks and Communications, Distributed Computing, Cloud Computing and Virtualization,. He is currently pursuing his Ph D in the field of virtualization of resources viz. Systems and network resources. His main focus is on Trust, Reliability and Fault Tolerant networks and systems for distributed and cloud environments.

Prof. Krishna Gopal is currently Dean Academic and Research at JIIT, Noida, India since 2011. He is Ph. D. from REC Kurukshetra Kurukshetra University Kurukshetra, India. He is having 45 years of teaching and R&D experience. He received his Bachelor, Master and PhD in Electronics engineering from IIT, Madras, REC Kurukshetra in 1966, 1972, 1979 respectively. He published more than 100 papers in different journals, conferences, patents etc. He is member of various professional bodies like: Life Member System Society of India, Indian Society for Technical Education, senior member of IEEE etc. His area of interest is Reliability and Fault Tolerant Networks and communication Systems.

Prof. J P Gupta has received his Ph D degree from University of Westminster, UK. He is currently the Director Emeritus (QA) at Hydrocarbons Education and Research Society, New Delhi, India. He is an academician having more than 35 years experience including Professor at IIT Roorkee, India, Vice Chancellor at JIIT Noida, India, Galgotia University, and Sharda University, India. He is author of more than 70 research papers published in International journals and conferences. He has received Commonwealth Fellowship and many more awards and memberships to his credit.