

# Efficient Resource Management Technique for Performance Improvement in Cloud Computing

Harvinder Singh\*

Research scholar, IKGPTU Kapurthala, Punjab, India  
harvinder9815653260@rediffmail.com

Dr. Anshu Bhasin

IKGPTU Kapurthala, India  
Dr.anshubhasin@ptu.ac.in

## Abstract

Cloud computing is growing technology which provide services to user's on affordable budget. Resource demand comes with different nature and create uncertainty situation for efficient resource matching and workload management on resources. Thus service providers' focus on suitable resource allocation technique for efficient resource matching based on resource availability and requirements. Such matching problems should be addressed using optimization approach. In this research paper we have presented an optimization approach using ant colony optimization (ACO) mechanism for efficient resource matching to minimize execution time. Our presented algorithm has been executed on cloud based simulation environment. The comparative analysis of results show that proposed algorithm performed better against existing algorithms and suitable for resource allocation in cloud to achieve quality of service (QoS) requirements and improve customer reliability.

**Keywords:** Cloud computing; performance; resource allocation; execution time; swarm intelligence.

## 1. Introduction

Cloud computing is providing dynamic service on public domain in form of infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS) based on quality of service (QoS) requirements. The cloud system motivates to organization to migrate from local commodity storage environment to on demand infrastructure which is based on pay-per-use model under service level agreement (SLA) rules. Thus, it avoids purchasing costly IT infrastructure and organizations use IT services on cost effective budget. As cloud is emerging platform to access information for different domains and energy consumption of resources impact on resource utilization and performance. Therefore, cloud service provider should manage efficient performance based on customer QoS expectations to improve user reliability and minimize execution time. So, cloud environment is becoming complex due to growing large scale applications. Performance can be improved based on QoS (execution time), if resources allocated as per demand from available resources. Resources allocation and management of resources direct impact on performance and consider as minimum execution time, which is difficult problem in clouds. As cloud users' demand for resource and expected results with limited time. Thus, service providers must ensure to provide effective cloud service as per application demand in the perspective of QoS with limited number of resources on minimum time. User demand have changed dynamically and difficult to maintain the load across the cloud resources as heterogeneity is there. We survey on existing research publications concerned to resource allocation problem which shown that presented techniques cannot be efficiently managed the cloud uncertainty problem where dynamic, heterogeneous and unpredictability exists. Therefore, an efficient resource allocation technique should be there based on QoS requirements to improve customer reliability and maximize profits. Resource allocation strategies impact on operational cost, utilization and energy consumption. To achieve effectively execution of task based on minimum execution time different resource management techniques used in cloud environment. Therefore, expected performance and QoS based results depends on resource management technique followed by resource matching procedure. "Cloud computing is a type of parallel and distributed system consisting of a dynamically

provisioned and presented as one or more unified computing resources based on SLA established through matching between the service provider and consumer” [1].

Efficient resource management support various complicated resource matching requirements and maintain expected performance and time. In cloud dynamic and heterogeneous environment optimization approach identify user demand based on resource availability to fulfill QoS requirements [2]. Efficient resource matching requires optimization process for identification of requirements. Hence, we motivate for optimization technique for efficiently identify resource requirements based on availability of resources and matching based on QoS. This technique further identifies runtime workload on resource to identify utilization and availability to provide expected performance and minimize execution time. Task resource allocation and management is generally resource uncertainty problem where user demand, resource discovery and monitoring, allocation of resource, optimization of resource utilization etc operations performed. As user demand for resources resource agent process started based on availability of resources to minimize execution time. In order to achieve high performance and QoS based requirements we have presented advanced ant colony optimization (AACO) algorithm for resource matching. We have implemented different resource management policies in cloud simulated environment for resource allocation based on QoS requirements.

In order to manage efficient performance on clouds VMs should be considered to gain the power of cloud resources. VMs perform virtually resource allocation to exploit physical machine and provide service based on different task context, which efficiently managed by resource management policy. This paper is organized as described sections: section 2 brief discussion on literature survey; section 3 presented resource matching algorithm; section 4 simulation based experimental work; section 5 results and analysis; section 6 conclusion and future work.

## 2. Literature Survey

In this section we briefly discuss on most relevant research work for resource management problems on clouds in the perspective of QoS requirements. To achieve the performance for expected results and efficiently management of resources different resource management techniques proposed for provider and user in the perspective of QoS. Resource (VM) allocation decision impact on performance, so resource allocation, mapping, scheduling should be efficient to achieve QoS requirements such as: cost, time, energy consumption, utilization, response time which further improve reliability and stability for effective cloud services. Existing survey [3-5] describes about resource management policies in the perspective efficiency, cost, reliability and utilization concerned to performance and QoS.

Mohammed et al. presented multiple redesigned VMs placement algorithm for efficient VM management to improve the performance and maximize resource utilization using clustering approach to consider VMs migration [6]. Sukhpal et al. presented pattern based resource provisioning technique in terms of QoS (execution cost and time) by analyzes and categories workloads based on requirements [7]. Masoud et al. presented energy aware method for resource allocation to improve performance and minimize energy consumption by appropriate allocation of resource [8]. Shahin et al. presented resource allocation model based on poison process where tasks access resources to reduce cost and improve utilization [9]. Fernando et al. presented probabilistic workload-aware dynamic resource allocation technique for optimizing resource cost especially for educational purpose (digital education) by identifying and classifying demand based on workload and QoS to avoid SLA violation [10]. Aarti et al. presented agent based automated service composition model for resource provisioning on clouds to reduce cost based on user requirement by appropriate resource assignment [11]. Zhen et al. proposed dynamic resource allocation to allocate resources based on demand and improved server utilization, and achieve good performance using skewness [12]. Jinn et al. proposed resources allocation method using differential evolution (DE) technique for load balancing and exploitation of resources to minimize makespan and cost and concentrate on processing time to improve performance [13]. Zhang et al. presented map reduced based fine grain resource aware scheduler to improve resource utilization and minimize makespan [14]. Daniel et al. presented dynamic resource allocation method (Nephele) to reduce execution cost and exploit cloud resources [15]. Anton et al. presented dynamic resource provisioning algorithm to minimize energy consumption and improve customer reliability [16]. Xingquan et al. presented PSO based resource allocation using self-adaptive PSO to provide the QoS based services [17]. Mohamed et al presented dynamic resource allocation to provide runtime reallocation for QoS but execution time is not considered [18]. Nattapong proposed optimization method for allocation of resources to tasks with optimal solution using PSO focus on computation cost and minimize makespan [19]. Dhinesh proposed scheduling algorithm for load balancing in system using honey bee behavior with focus on task waiting time when tasks are independent on VMs [20]. Jignesh proposed grouping of tasks before resource allocation to reduce communication cost and computation cost [21]. Elina et al. proposed optimization based algorithm for delivering reliable and

scalable services based on custom VMs launched in appropriate hosts using ACO [22]. Peng et al. presented resource allocation strategy for application oriented allocation using VM placement strategy to minimize execution time and performance improvement [23]. Parnia et al. presented market based combinatorial double auction resource allocation model for appropriate resource allocation to provide user satisfaction and minimize cost [24]. Manish et al. presented dynamic resource demand prediction and allocation framework for efficient VM allocation using classification and allocated VMs to hosts as per scaling requirements to achieve performance [25]. Annena et al. presented load aware resource allocation strategy for efficient resource management using cuckoo-search algorithm to minimize makespan and cost [26].

### 3. Our Presented Resource Management Algorithm

In presented resource allocation model user submitted number of tasks as represented with figure 1. Resource agent collected and evaluated user demand based on resource requirement and resource availability to find minimum execution time. The resource agent received user tasks and consolidates requirements before forward to VMs. Cloud user expectations that result must be based on QoS requirement such as minimum execution time. Sub-optimal solutions generated based on tasks demand and calculating using heuristic information, known as computation time for task execution. Resource information represented the availability and configuration of resources. To evaluate user demand the component suboptimal solution represented computation time and generated number of solutions as per user defined criteria. Sub-optimal solutions generated based on ACO control parameters using discovery process. After random generation steps over predictable value found cost effective solution. Further, sub-optimal solution negotiated for optimal solution which provided based on QoS requirements to provide user satisfaction. Allocation has been made based on optimal solution to minimize execution time and improve performance. Further, mapping of resource has done based on allocation decisions. The alert message sent by task execution to resource information regarding resource availability. Task execution process generated based on QoS parameters and resource information updated. The resource availability information updated based resource dispatched for scheduling. Task execution executed tasks and update resource information to update resource available pool for allocation. Further, tasks and VMs status updated and resource management policy improved resource utilization and performance and minimize execution time.

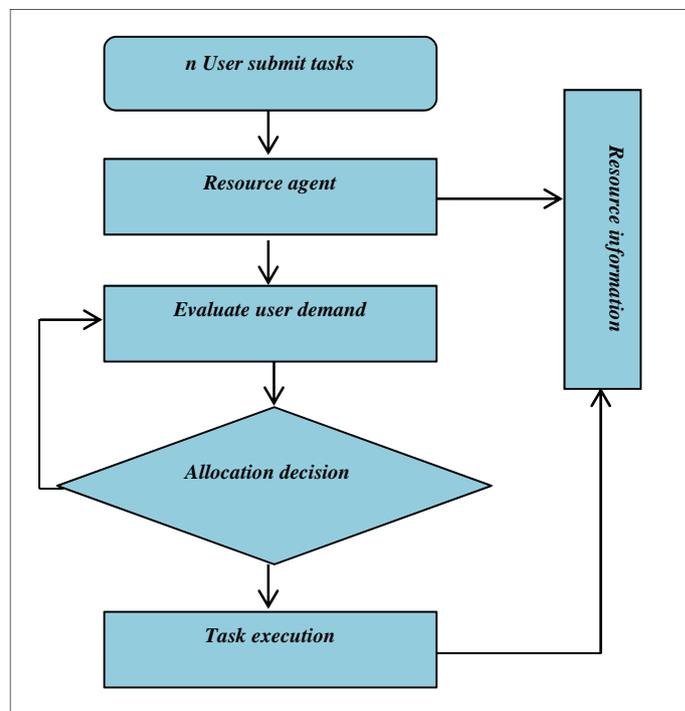


Fig. 1 Flowchart of resource management technique

The presented resource management technique performed based on allocation of VMs. The proposed resource allocation algorithm allocated resource based on negotiation process based on requirements to minimize execution time. Our presented algorithm pseudo code as follows:

```

T:      user tasks
R:      number of available resources
TimeRC: computation time on resource R for task T
MinTime: minimum computation time known as best match
Step 1: Input: Tasks T1, T2, T3, Tn from users
Step 2: Check the availability of resources
Step 3: resource ready list R1, R2, R3.. Rn
/* check computation time on available resources */
Step 4: TimeRC = for R1 ∈ Rn / T1 ∈ Tn
/* Find minimum computation time on resource */
Step 5: MinTime(TimeRC)
/* Find best resource for allocation */
Step 6: R1T3, R2, T2, R3T1, RnTn
/* update resource list and dynamic positioning */

```

In our presented algorithm user submitted number of tasks and information about resource requirement is collected through optimization process to improve performance and minimize execution time. User demand is dynamic and heterogeneous thus allocation considered using optimal solution. Resource availability affected the allocation decision and performance. So, dynamic resource pool performed the information about resources. Before resource allocation, the resource agent collected the information about VMs as well as their configuration and present workload information about resources. The workload information associated with the computation power available of the CPU to consider as under demand and current utilization of resources. The resource agent collects the computation time of tasks on available resources by visiting tasks randomly on VMs. This information act as swarm agents which moved for the best solution based on previous trails (random generated solutions). The computation time information is stored in *Time<sub>RC</sub>*; the comparison of stored information with each resource for tasks. Further, to filter minimum time *MinTime(Time<sub>RC</sub>)* operation performed which have taken less execution time from all generated solutions and known as optimal solution. The allocation policy in this way improved the workload management and utilization of resource and minimized execution time.

In cloud uncertainty environment for efficient management of resources allocation problem performed with swarm optimization approach which improved the probability to search most suitable resource for task with minimum execution time. Swarm technique is based on nature inspired and artificial intelligence system which consists of autonomous agent with collective behavior using decentralized system [27]. The agents continue moving in random search which follow some rules during the process. The moving process is trying to find best solution without central command. The resource management problem requires finding most suitable match and swarm technique ability is to inspire researchers to practice the experimental work for resource allocation is interesting approach. Optimization process has taken time due to number of sub-optimal solution generation process, calculation of execution time for task execution on resource. Sub-optimal solutions generated based on different random parameters to cover probabilities as low cost solution found within system. This core step performs before resource allocation to user, QoS concerns countable to minimize time. Sub-optimal solutions mark as computation cost and time as pheromone value lay down on ants' route. As max step reached collective communication generate optimal solution, suitable for user task and resource, which provide customer and provider satisfaction, results in low cost and improve cloud resource user ability. Solution process continues till all virtually mapping solutions over. Further resource management performed by resource agent based on selected optimal solutions. VM scheduler executes tasks and resource information is updated dynamically to minimized execution time inspired by ants distributed and dynamic nature.

#### 4. Simulation based Experimental Work

The presented resource allocation algorithm for efficient resource management using ant colony optimization has implemented on cloud simulated environment workflowsim, extended version of cloudsims which provide cloud environment, Data center, Host, VMs configuration, RAM, Bw, MIPS, File size, as essential components for cloud environment [28]. Moreover, workflowsim provide scheduling environment where existing resource management policies can be performed, and custom scheduling policy can be added to manage different requirement. Workflowsim is advanced tool for algorithm experiments in cloud computing which inclusion of communication

overheads, support clustering and advanced features for supporting fault tolerance in distributed cloud environment. Workflowsim further provide platform to execute and simulate proposed resource management policy to evaluate efficiency. So, this simulation environment fulfills our experimental work requirement to test resource utilization and performance. The provided results evaluate in the perspective of cost and time. The presented resource management policy achieved the users' QoS requirements and improved performance.

Our resource management algorithm experiment conducted with varies number of VMs and workflow based Montage data set has been taken as workload. The Montage application created by the NASA/IPAC has been represented as a workflow that can be executed in cloud and Grid environments such as the TeraGrid [29]. The proposed algorithms performance has been measured based on execution time to compare the performance with other existing resource management policies.

### 5 Results and Analysis

The presented algorithms implemented using workflowsim simulation framework that enables modeling, simulation and experimentation of task scheduling in cloud environment and provide varies configuration of cloud resources for services. The simulation environment consists on one Data Center with VMs 5 to 50 for varies cloud environment. The presented algorithm performance tested on different number of VMs using Data set as Montage with 100 cloudlets. The algorithms performance measure based on execution time metrics for execution the cloudlets. The graphical results as shown in figure 2 as the number of VMs increases AACO algorithm performance has been better compared to FCFS, MCT and ACO and has taken less time for execution of tasks.

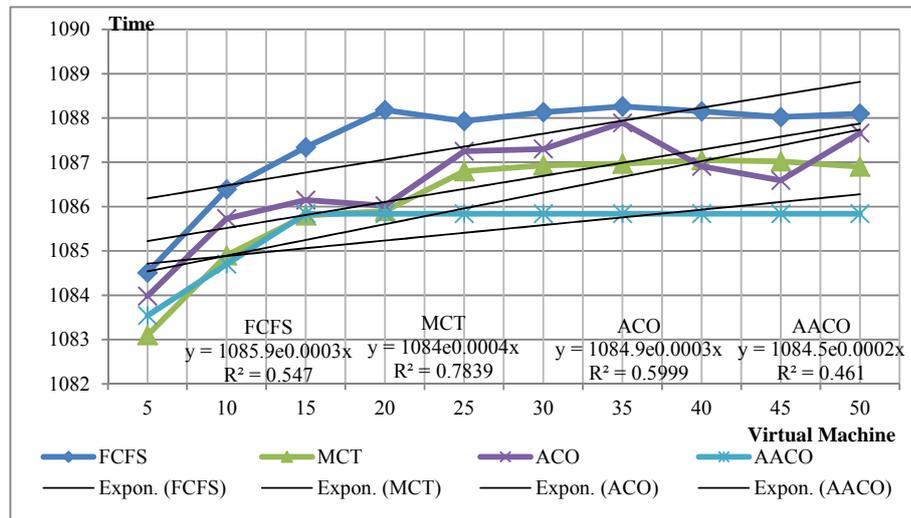


Fig. 2 Graphical data represent varies number of VMs execution of cloudlets 100 and computation cost

The results are plotted against execution time; here R2 value calculated using exponential curve. The value near to 1 shows exponential growths in all the values where in the other hand value near to 0 show exponential decrease. The figure 2 represented with respect to time for 100 tasks on set of VMs 5-50. It is observed from the table (table 1) that experiment is executed for FCFS, MCT, ACO and AACO (Designed Policy). The results of the resource management techniques have compared in terms of R2 value and it was observed from graph (figure 2) that designed algorithm (AACO) has worked best by taking minimum time compared to the existing policies.

The Table I shows the R2 value for Time of executing 100 Tasks, here it has observed that designed policy executed 100 tasks with lesser time compared to existing polices as the value of R2 is near to 0.

#### Observations for Time for 100 Tasks

15.72 %, 41.12 %, 23.04 % of improvement in time compared to FCFS, MCT and ACO which show that presented AACO has efficiently allocated resources. So, based on observations AACO considered as efficient resource management policy for performance improvement in cloud computing.

TABLE I. R2Value for execution of 100 tasks

| R <sup>2</sup> | Algorithm |       |       |       |
|----------------|-----------|-------|-------|-------|
|                | FCFS      | MCT   | ACO   | AACO  |
|                | 0.547     | 0.783 | 0.599 | 0.461 |

## 6. Conclusion and Future work

Cloud computing is becoming more beneficial for consumers which provide service based on user demand and performance is considered important parameters for both sides. Cloud resource management problem depends upon resource agent which provided service using different resource allocation strategies. The presented resource management policy has been used ACO technique as optimization for efficient resource allocation and management to consider QoS requirements. Cloud simulated environment configured with different set of VMs 5-50. The comparison between policies demonstrated that presented AACO technique outperformed compared to FCFS, MCT and ACO. So, our policy achieved users' QoS requirements and provided the state of art for using cloud as a service. As cloud computing trend is growing the complexity in resource demand will become more complex. Therefore, efficient technique would be developed to manage heterogeneous, dynamic, and uncertainty in cloud environment. So, our plan is to extend the work for resource utilization using optimization technique to fulfill users' QoS requirements. This technique would fulfill consumer and service provider QoS requirements by efficiently manage cloud resources.

## References

- [1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic, (2009): Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5<sup>th</sup> utility. *ELSEVIER Future Generation Computer Systems*, Vol. **25**, no.6, pp. 59-616.
- [2] S. Sukhpal, C. Inderveer, "Resource provisioning and scheduling in clouds:QoS perspective," *Springer Journal of Supercomputing*, DOI 10.1007/s11227-016-1626-x,.
- [3] Z. Jiangtao, H. Hejiao, W. Xuan, (2016): Resource provision algorithms in cloud computing: A survey. *ELSEVIER journal of Network and Computer Applications*, **64**, pp. 23-42.
- [4] S. Mustafa, N. Babar, H. Amir, R.K. Atta, A. M. Sajjad, (2015): Resource management in cloud computing: Taxonomy,prospects, and challenges. *ELSEVIER Computers and Electrical Engineering*, vol. **47**, pp.186–203.
- [5] I. Pietri, S. Rizos, (2016): Mapping Virtual Machines onto Physical Machines in Cloud Computing: A Survey. *ACM Computing Survey*, vol. **49**(3), pp. 1-30.
- [6] R.C. Mohammed, R.M. Mohammed, M.R. Rashedur, "Implementation and performance analysis of various placement strategies in CloudSim," *Springer Journal of Cloud Computing: Advanes Sys. & Appl.* (2015). DOI 10.1186/s13677-015-0045-5
- [7] S. Sukhpal, C. Inderveer, (2015): Q-aware: Quality of service based cloud resource provisioning. *ELSEVIER Computers and Electrical Engg.*, **42**, pp. 138-160.
- [8] N. Masoud, K. Ronak, (2016): Energy efficient and latency optimized media resource allocation. *Emerald International Journal of Web Information Sys.*, vol **12**(1), pp. 2-17.
- [9] V. Shahin, M.A. Mustafa, Q. Dongyn, (2015): Modeling of the resource allocation in cloud computing centers. *ELSEVIER Computer Networks*, vol. **91**, pp. 453-470.
- [10] K. Fernando, D.A. Marcos, C.Carlos, A.S.N.Marco, (2015): Optimising resource costs of cloud computing for education. *ELSEVIER Future Gen. Comp. Sys.* Vol. **55**, pp. 473-479.
- [11] S. Aarti, J. Dimple, M. Manisha, (2015): A novel agent based autonomous and service composition framework for cost optimization of resource provisioning in cloud computing. *ELSEVIER Journal of King Saud University Computer and Information Sci.*, doi.org/10.1016/j.jksuci.2015.09.001.
- [12] X. Zhen, S. Weijia and C. Qi, (2013):Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment. *IEEE Tran. Parallel and Distributed*, vol. **24**, no. 6, pp. 1107-1116.
- [13] T. Jinn-Tsong, F. jia-Cen and C. Jyh-Horng, (2013): Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. *ELSEVIER, Computers & Operations Research*, vol. **40**, no.12, pp. 3045-3055.
- [14] Zhang Qi, Z. Mohamed, Y. Yuke, B. Raouf and W. Bernard, (2015): PRISM- Fine-Grained Resource-Aware Scheduling for MapReduce. *IEEE Trans. Cloud Computing*, vol. **3**, no. 2, pp. 182-194.
- [15] W. Daniel and K. Odej, (2011): Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud. *IEEE Trans. Parallel and Distributed*, vol. **22**, no. 6, pp. 985-997.
- [16] B. Anton, A. Jemal, R. Buyya, (2012): Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. *ELSEVIER, Future Generation Computer System*, vol. **28**, pp. 755-768.
- [17] Z. Xingquan, Z. Guoxiang and T. Wei, (2014): Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud. *IEEE Trans. Automation Sci. and Engg.*, vol. **11**, no.2, pp. 564-573.
- [18] S. Mohamed, J. Manar, S. Abdullah and O. Abdelkader, (2013): Resource Allocation in a Network-Based Cloud Computing Environment: Design Challenges. *IEEE Communications Magazine, Cloud Network and Communications*, pp. 46-52.
- [19] N. Nuttapon, S. Booncharoen and A. Tiranee, (2012): Cost Optimization in Cloud Provisioning using Particle Swarm Optimization. *IEEE 9<sup>th</sup> Int'l Electrical Engg./Electronics, Comp., Telecommunications and Information Technology*.
- [20] L.D. Dinesh and K.Venkata, (2013): Honey bee behaviour inspired load balancing of tasks in cloud computing environments," *ELSEVIER Applied Soft Computing*, Vol. **13**, pp. 2292-2303.
- [21] L.Jignesh, A.B. Hitesh, (2013): An Approach to Optimized Resource Scheduling using Task Grouping in Cloud. *IJARCSSE*, vol. **3**, no. 9, pp.1-6.

- [22] P. Elina, M. Cristian, G.G. Carlos, (2015): Balancing throughput and response time in online scientific Clouds via Ant Colony Optimization (SP2013/2013/00006). ELSEVIER, Adv. In Engg Soft, vol. **84**, pp. 31-47.
- [23] Jun-jie Peng, Xiao-fei Zhi, Xiao-lan, Xie, (2016): Application type based resource allocation strategy in cloud. ELSEVIER Microprocessors and Microsystems, pp.1-7. <http://dx.doi.org/10.1016/j.micpro.2016.09.014>
- [24] S. Parnia, T. Youness, M. Muriati, (2016): A combinatorial double auction resource allocation model in cloud computing. ELSEVIER Information Sciences, Vol. **35**, pp. 201-216. <http://dx.doi.org/10.1016/j.ins.2014.02.008>
- [25] V. Manish, G.R. Gangadharan, N.C. Narendra, R. Vadlamani, V. Inamdar, L. Ramachandran, N.C. Rodrigo, R. Buyya, (2016): Dynamic resource demand prediction and allocation in multi-tenant service clouds. WILEY Concurrency and Computation Practice and Exp.. DOI.10.1002
- [26] A.A. Aneena, L.J. Divya, , (2016): An Efficient Resource Management For Prioritized Users In Cloud Environment Using Cockoo Search Algorithm. ELSEVIER Procedia Technology **25**, RAEREST, pp. 341-348.
- [27] P. Elina, M. Cristian, G.G. Carlos, (2014): Distributed job scheduling based on Swarm Intelligence: A survey. *ELSEVIER, Computers and Electrical. Engg.*, vol.**40**, pp. 252-269.
- [28] C. Weiwei, D. Ewa, (2012): WorkflowSim: A Toolkit for Simulating Scientific Workflow in Distributed Environments," 8<sup>th</sup> IEEE Intl. Conf, on eSci 2012, Chicago.
- [29] <http://pegasus.isi.edu/schema/DAX>