International Journal of Pure and Applied Mathematics
Volume 117 No. 21 2017, 591-604
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

Resource Scheduling for Private Cloud Using Cloud Platform Adaptive Genetic Algorithm (CPA-GA)

[1]A. AnushaPriya, [2]R. Gunasundari and [3]K. Appathurai

[1]Department of Computer Science,

Karpagam University,

Karpagam Academy of Higher Education,

Coimbatore.

[2]Department of Information Technology,

Karpagam University,

Karpagam Academy of Higher Education,

Coimbatore.

[3]Department of IST,

Sur University College,

Oman.

Abstract

There are many thrust research areas in the cloud computing arena. Organization makes use of its own cloud environment and it is called private cloud. The resources such as computer, storage and network play a significant role in providing cloud computing services. Such resources need to be used in an effective manner and scheduling of such resources is an ever demanding research paradigm in such private cloud environment. This research work presents cloud platform adaptive genetic algorithm in order to effectively schedule and allocate the resources. Modifications in conventional GA are made in order to adapt for private cloud environment. In this CPA-GA, a cloud adaptive encoding scheme is employed. Earliest Release Time (ERT) and Earliest Completion Time (ECT) are used during population initialization. Genetic operators are used and two local search strategies are performed for obtaining better outcome. Performance metrics such as resource utilization, average response time and makespan are taken into account for evaluating the performance of CPA-GA. Simulation results prove that CPA-GA outperforms in terms of the obtained metrics.

Key Words: Cloud computing, resource scheduling, virtual private cloud, resource allocation, average response time, makespan, resource utilization, genetic algorithm.

1. Introduction

Cloud computing is on par with the recent computing technologies that helps an extraordinarychangethe way organizations face their computing requirements. With the help of immenseamalgamation of prevailing computing servers (called as data centers) and colossal data storage space, cloud systems distribute three disposition models namely IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service) [1]. The applications that are cloud-centred (i.e. applications using cloud deployment models as a platform) are globally used, that results in melodramaticprogress of cloud resource convention. Hence there is a vast scope of resource scheduling and resource allocation of such mammoth amount of computing tasks and distributing the data files. Resource allocation and job scheduling are the thrust research area in the field of cloud computing [2]. This research paradigm is the due course of resource availability that is in the cloud data center. Hence there is a vast research scope to employ right mechanisms that will be capable enough to allocate the resources and schedule the jobs to the resources.

Cloud systems are connected to jobs via a scheduling podium. This podium is usually called as scheduler that obtains application from the user, analyzesthe application and allotting it to certain required computing resource(s). In a real-time scenario, analyzingscheduling task has a fairlytrivialextent when linked with the total job execution time [3]. On the other hand, cloud schedulers probably need composite analysis subsequently they administer a vast number of things namely bandwidth, instance types, job's computing requirements, data file sizes, and dependencies among others. With the huge number of conceivable solutions this cloud computing environment incites, the scheduling of workflows cascades into the type of an NP-hard problem which means that it is difficult to be get solved during available polynomial time by making use of current computing schemes [4]. For the above said outlines, this research work aims in proposing cloud platform adaptive genetic algorithm for resource scheduling scheme for private cloud environment.

2. Related Works

In this section, related works are studied and presented briefly. Generally the cloud applications are mentioned as BoT (Bag of Tasks) and in research terminology; they are mentioned as DAGs (Direct Acyclic Graph). In common, BoTs contains parallel jobs that are free from one another in terms of data dependencies [10,11]. But as far as DAGs are concerned, the selection of VM pool size is (i) driven by a monetary cost constraint, (ii) selected by the user or (iii) computed by the scheduler. As far as this review of literatures are concerned, utmost all DAG schedulers does not have effective job dependency patterns. In the literature authored by Kloh et al. [6] a bi-criteria method to schedule DAGs in cloud environments with monetary cost constraint guidance to select a number of resources is proposed. Their method optimizes two

variables out of runtime, cost and/or reliability. The process starts with the application owner selecting two objective variables, then incorporating them into well-known scheduling mechanisms including bi-criteria scheduling [12], dynamic scheduling [13], fault tolerance policies [14], cost-based scheduling [15], multiple QoS constrained schedule strategy [16], and scheduling decisions based on service level agreements [17]. Through experimentation, the authors claim that their bi-criteria scheduler is better to Join the Shortest Queue (JSQ) method mentioned in the literature [18].

The research work proposed by Achar et al. [9] generated a scheduling algorithm orientated for BoT applications with VM pool size selection said by the user. Their approach mainly prioritizes jobs and VMs based on MIPS (Million Instructions per Second). Then it groups jobs and selects the best group of VMs to execute them. This algorithm obtains high resource utilization with low execution times compared to FCFS (First Come First Serve). On the other hand, the authors used several VMs in their simulations. Furthermore, their algorithm does not consider network behavior with large data file sizes.

Zhang et al. [8] modelled a scheduler based on the Ordinal Optimization (OO) method that accomplishes scientific workflows on a fixed number of resources. The OO method's objective is to diminution the scheduling time overhead by reducing the solution space. Hence, the authors modified the OO method originally developed for automated systems. From their experiments the authors claims that their modified OO reduced scheduling overhead compared with the Monte Carlo method. From this literature it is inferred that their modified OO neither scales up nor down the number of resources nor provides user guidance for this selection.

Deng et al. [5] developed a linear programming algorithm to plot applications to a static pool of resources/VMs with the objective of reducing monetary cost and response time. The nature of the scheduling decisions, based on arrival time, identifies this approach as a BoT scheduler. Granting the scheduler is capable enough to leverage a different number of resources; theirproposed work lacks a precise policy to select the number of VMs and/or to limit the number of resources by a monetary constraint. Additionally, in their work the authors didn't deliberate applications with interdependent jobs.

Moschakis et al. [7] conducted a study for the execution of jobs on a different number of resources/VMs arranged on the Amazon Elastic Compute Cloud (EC2). Their dynamic scheduling approach considers BoT applications incoming with exponential distribution time. In their experiments they have evaluated AFCFS (Adaptive First Come First Serve) and LJFS (Largest Job First Serve) methods. From their experiments, it has been inferred that AFCFS executes jobs as soon as they arrive while the LJFS executes jobs previously prioritized according to their computing requirement demands. Nevertheless, this approach lacks an analysis to handle applications with job dependencies; for this reason, this scheduler is incongruous.

Tsakalozos et al. [21] modelled a scheduler with a malleablecollection of VMs that equilibriums cloud revenue and user inexpensive. FSV (Flexible Selection of VMs), has been chosen by the authors that calculates the number of VMs required to implement an application based on its computing requirements. At one part, the users' aim is to run their application outlaying their budget. On the other part, cloud service providers portray to exploit theirincome. Conversely, this will be done by only balancing the number of VMs based on computing demands, i.e. it uses the maximum number of VMs without considering monetary cost optimization and a consequence users spend their entire budget without contemplating options with a different number of VMs.

3. Proposed Work

Preliminaries

There are certain preliminaries and assumptions for this research work. It is considered that one distributed application has certain resources that need to be scheduled at the time of file sharing. This file sharing mechanism is denoted by Directed Acyclic Graph (DAG). The vertices of DAG denote the segmented resources of the cloud user. It is also considered that the edges of DAG denote the linkage between the resources. This is mathematically defined as $G=(N,E)$ where 'N' consists of a collection of 'n' cloud users. $ClUsrs=u_1,u_2,u_3,...,u_n$ and a group of 'E' edges denoted as 'e'. It has been taken that $res=res_1,res_2,res_3,...,res_n$ that represents the type of resources. The resources considered in this research are memory which is nothing but the storage space that is available in the cloud data centers).

Non – Static Resource Scheduling (NSRS)

The goal of NSRS method is to lessen the overall waiting time and to extend the resource scheduling efficacy for each cloud user resource request in private cloud milieu. This NSRS method has several segments.

- Requests of Cloud Users.
- Cloud User Interface (ClUsrInt).
- Virtual Machine Grid (VM-G) Agents.
  - Virtual Machine Bandwidth Grid Agent
  - Virtual Machine Memory Grid Agent
  - Virtual Machine CPU Grid Agent
- Resource Manager

Depending upon the requests raised by cloud users, the request arrival rate is denoted by λ; the arrival rate of the i[th] virtual machine in the grid 'j' is mentioned as λ_{ij}. The service rate is expressed mathematically as μ_{ij}. The average queuing time of user request is denoted as

$$\mathrm{Re}\,qQT=\frac{\rho_{ij}}{\mu_{ij}\,(1-\rho_{ij})} \quad \ldots (1)$$

In the above equation (1) the value of ρ_{ij} is computed using $\rho_{ij} = \frac{\lambda_{ij}}{\mu_{ij}}$ .The main task of the ClUsrInt is to obtain the cloud users request, estimate the resource requirement, allot and schedule them to the appropriate VM-G depending upon the requests made by the cloud user, getting the execution results and sending back the same to the ClUsr. VM-G is formed by the set of VMs that will schedule same resource during file sharing. Each VM will have its own resource manager shortly termed as RM. The RM will have the responsibility of resource management such as bandwidth, memory and CPU. The estimation of requirements is done using the subsequent moving average. The value of first moving average is obtained and then the value of second moving average is measured at time 'T' time of the 'ath' user which is mathematically represented in equation (2).

$$MAvg_{T}^{1}(a) = \frac{x_T(a) + x_{T-1}(a) + ... + x_{T-(N-1)}(a)}{N} \quad ...(2)$$

Resource Scheduling using Cloud Platform Adaptive Genetic Algorithm (CPA-GA)

Cloud platform adaptive genetic algorithm has been proposed in this research. This resource scheduling problem is considered as NP-hard problem. From several literatures it is obvious that GA will perform well in solving this NP-hard problem specifically in the heterogeneous circumstances than the old-fashioned optimization algorithms. There are certain key elements that make CPA-GA different from conventional GA. Primarily; the individuals are segregated into a number of ranks at every generation. This is carried out by making use of swift sorting method. At the next stage, a criteria-free sharing method is used. This criteria-free sharing method is carried out through crowding distance of individuals in the same rank. After that, selection operation is performed based on rank and crowding distance values of individuals. At last, an elitism mechanism is employed to increase the convergence.

1. Individual Representation of CPA-GA

There are two discrete decisions, namely, resource allocation and resource scheduling, that needs to be performed in parallel fashion. The allocation sub-problem is concerned with the assignment of jobs on the VMs. On the other hand, resource scheduling sub-problem is to govern the sequence of jobs on every VM. Hence there is a need for exclusive encoding scheme in order to represent an individual in the population. Hence in this research work, a separator is used to split an individual into two parts that represents the scheduling sequence of jobs on two VMs.

2. Population Initialization of CPA-GA

In this research work, cloud environment adaptive heuristic method is presented for generating an individual with higher quality. In this method, earliest release time (ERT) heuristic in which the jobs with the earlier releasetime have higher process priority and earliest completion time (ECT) heuristic, where the jobs

with theearliest completion time have higher priority to process, are combined together. The algorithm is given below.

```
Begin
sort all jobs in J with an ascending order via the release time;
set t1=t2=t3=0;
set J1=J2=∅;
while J!= ∅
select the first job j in J;
t'1=max{ri, t1}+pj1;
t'2=max{t1, t2}+pj2;
t'3=max{ri, t3}+pj3;
if t2<=t'1+wj max then
        J1=J1U{j}; t1=t'1; t2=t'2;
else if t'2+p'<=t'3 then
        J1=J1U{j}; t1=t'1; t2=t'2+p';
else
        J2=J2U{j}; t3=t'3;
end if
remove j from J;
end while
end
```

Notation and Descriptions

J: set of all jobs;
J1: set of jobs in VM - 1;
J2: set of jobs in VM - 2;
t_m: earliest available time of machine m, m={1, 2, 3};
t'_m: temporary value of tm, m={1, 2, 3};
p_{jm}: the process time of job j in machine m, m={1, 2, 3}.

For increasing the eminence of the initial population, one half of individuals are generated with the help of reverse mutation operation upon an elitist individual. The left-side and right-side individuals are generated randomly in order to maintain the diversity of the initial population.

3. Genetic Operations of CPA-GA

Binary tournament selection is employed in CPA-GA. At first, two individuals are chosen in a random fashion from the current population. Once after choosing, the strong individual is stored into the mating pool. When the two individuals are not strong, they are both stored onto the mating pool. After N individuals, whereN is population size, are chosen, the crossover operation is performed till the existence of individuals in the mating pool.

The crossover operation is given below:

1. Select parent individual 1, and take the genes locating in the left of separator into the offspring individual 1;
2. Delete the selected genes from the parent individual 2, and add the left

genes into the empty positions of offspring individual 1. The offspring individual 2 is also generated in the same way. Every recently produced offspring individuals by crossover will endure mutation operation with thee mutation probability of 0.1. At first, two dissimilar positions in an individual are chosen randomly. Next, one mutation way is chosen from swapping with the same mutation probability of 0.1. At last, the mutation operation is executed.

4. Local Search of CPA-GA

In CPA-GA, two unalike local move strategies are employed. The first one is mentioned as repair strategy. In this first strategy, the delayed jobs that are assigned to VM-1 are re-allocated to VM-2. Apparently, this strategy will enhance the quality of individual which results in lessening the number of delayed jobs. The next strategy is a random strategy that interchanges, inserts or reverses mutation method upon the individual in random fashion. This random moving gets stopped when the maximal iteration number is reached. In this research maximum number of iteration is set to 500.

4. Simulation Model

The simulation model for this research consists of private cloud environment that has two varieties of computing nodes namely AMD and Intel. As far as AMD computing environment is concerned, it has 8 core processors, RAM capacity of 32 GB and secondary storage memory capacity of 100 GB. As far as Intel computing environment is concerned, it has 12 core processors, RAM capacity of 32 GB and a secondary storage memory capacity of 200 GB. AMD computing environment is capable enough to create maximum two VMs without any performance deprivation. The elapsed time for VMs to create in the AMD environment is 120 seconds and 2 seconds to power it off. In the Intel computing environment it has the capacity to create maximum of three VMs. The elapsed time to create VM is 60 seconds and 1 second to power off. In order to evaluate the performance of the proposed scheduling environment 64 nodes are created (32 in AMD environment and 32 in Intel environment). Totally 1000 jobs are allocated for simulation. The metrics namely, resource utilization, average response time and makespan are considered for performance evaluation.

5. Results and Discussions

The performance results of resource utilization are presented in Fig.1 and Fig.2. The proposed CPA-GA is compared with the existing BaRRS [20] mechanism. At first AMD VMs are evaluated. 32 nodes were created during the simulation. From the results shown in Fig.1, it is observed that the existing BaRRS [20] mechanism made use of the resources at the maximum of 84.78% and at the minimum of 80.04%. At the same time, the proposed CPA-GA made use of the resources at the maximum of 91.97% and at the minimum of 88.05%. Hence it is evident that the proposed CPA-GA mechanism outperforms than that of

BaRRS in terms of resource utilization in AMD VMs. The simulations are also conducted with Intel VMs. 32 nodes were created. From the results shown in Fig.2, it is observed that the existing BaRRS [20] mechanism made use of the resources at the maximum of 86.98% and at the minimum of 83.28%. At the same time, the proposed CPA-GA made use of the resources at the maximum of 93.69% and at the minimum of 91.09%. Hence it is obvious that the proposed CPA-GA mechanism outperforms than that of BaRRS in terms of resource utilization in Intel VMs.

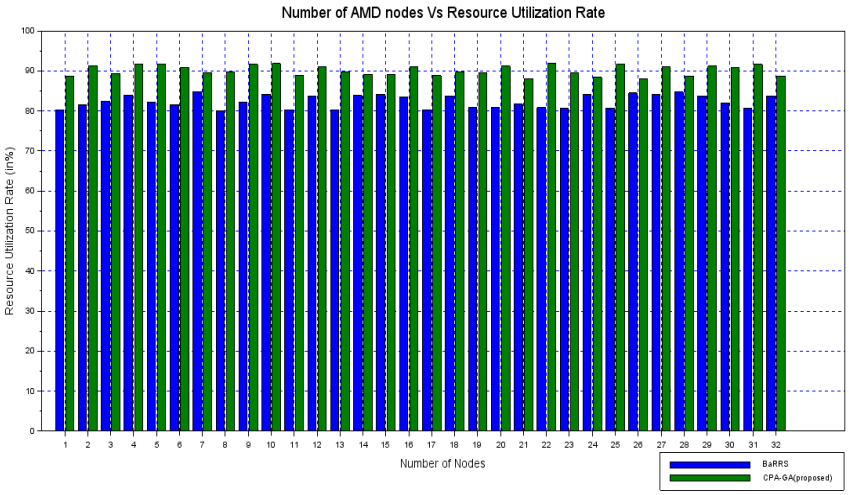

Figure 1: Number of AMD Nodes Vs Resource Utilization Rate

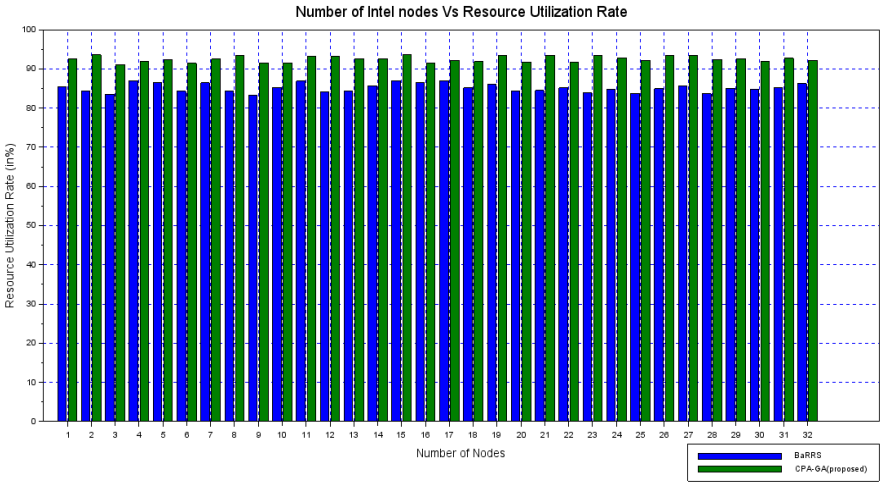

Figure 2: Number of Intel Nodes Vs Resource Utilization Rate

Next, the performance result of average response time is shown in Fig.3. The proposed CPA-GA has been compared with the existing BaRRS [20] mechanism. From the results depicted in Fig.3, it is observed that the average response time of BaRRS [20] for AMD VMs is 105.8 seconds and for Intel VMs is 99.3 seconds. Hence it is understandable that the proposed CPA-GA

mechanism outperforms than that of BaRRS in terms of average response time in both AMD VMs and Intel VMs. Subsequently, the performance result of makespan is shown in Fig.4 and Fig.5. The proposed CPA-GA has been compared with the existing BaRRS [20] mechanism.

From the results shown in Fig.4, it is observed that the makespan time of BaRRS [20] for 32 AMD VMs is more when compared with the makespan time of the proposed CPA-GA. From the results shown in Fig.4, it is clear that the makespan time of BaRRS [20] for 32 Intel VMs is more when compared with the makespan time of the proposed CPA-GA. From the Fig.4 and Fig.5, it is inferred that the proposed CPA-GA consumes less makespan to execute the given 1000 jobs.

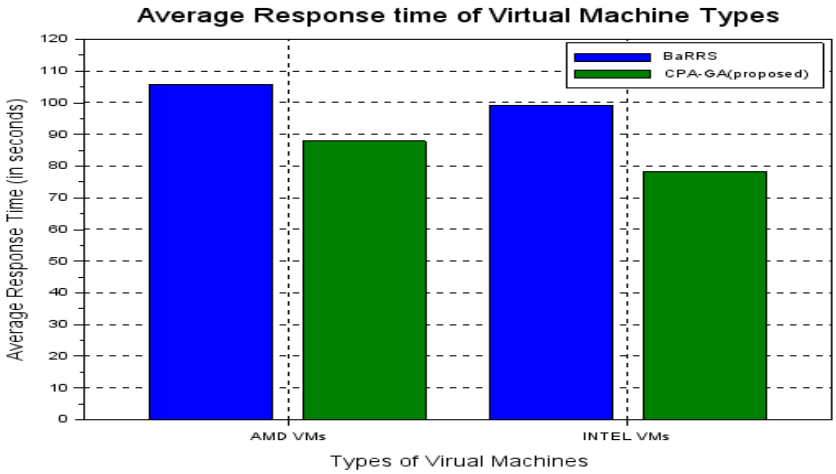

Figure 3: Performance Evaluation – Average Response Time

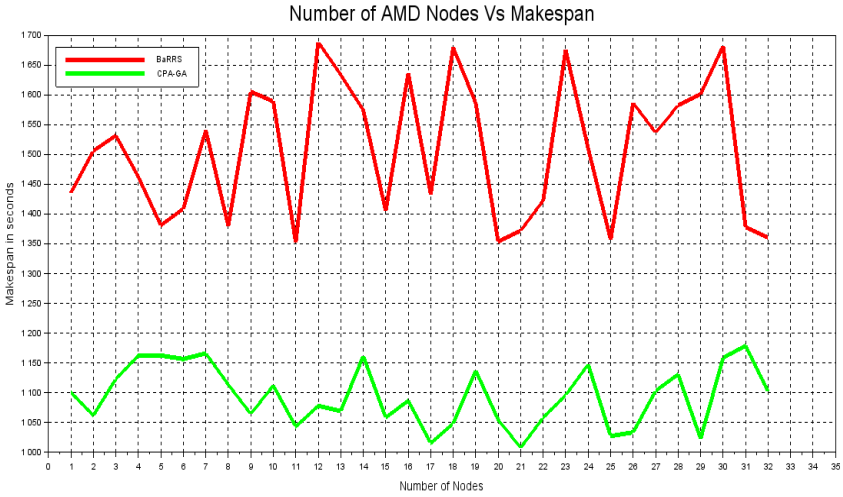

Figure 4: Number of AMD Nodes Vs Makespan

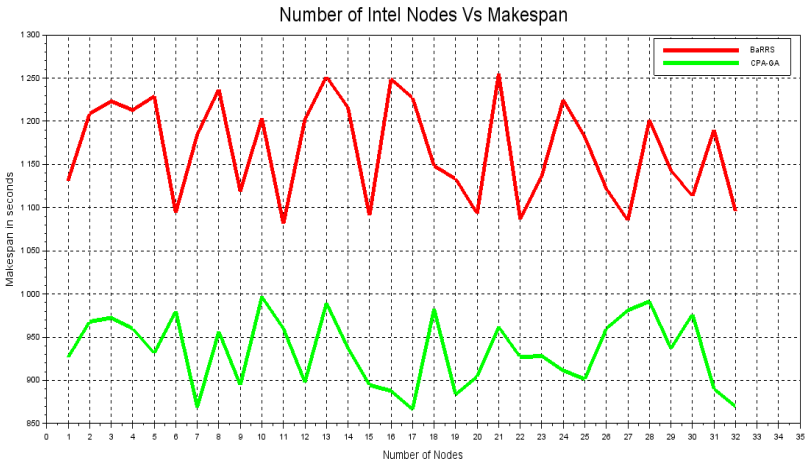

Figure 5: Number of Intel Nodes Vs Makespan

6. Conclusions

This research work aims in providing an effective solution for resource scheduling problem in private cloud environment named CPA-GA. In CPA-GA, a cloud adaptive encoding scheme is employed. Earliest release time (ERT) and earliest completion time (ECT) are used during population initialization. Genetic operators are used and two local search strategies are performed for obtaining better outcome. Performance metrics such as resource utilization, average response time and makespan are taken into account for evaluating the performance of CPA-GA. Results prove that CPA-GA performs than that of BaRRS [20] scheduling strategy in terms of increased resource utilization and lesser makespan with reduced average response time.

References

[1] Iosup A., Ostermann S., Yigitbasi M.N., Prodan R., Fahringer T., Epema D.H., Performance analysis of cloud computing services for many-tasks scientific computing, IEEE Transactions on Parallel and Distributed Systems (2011), 931-945.

[2] Illumina. Available: https://www.illumina.com/

[3] (2013, July) IEEE SPECTRUM.

[4] Martini B., Choo K.K.R., Cloud storage forensics: own Cloud as a case study, Digital Investigation (2013), 287-299.

[5] Deng L., Yu Q., Peng J., Adaptive scheduling strategies for cloud-based resource infrastructures, In Security and Communication Networks (2012), 1102-1111.

[6] Kloh H., Schulze B., Pinto R., Mury A., A bi-criteria scheduling process with CoS support on grids and clouds, Concurrency and Computation: Practice and Experience (2012), 1443-1460.

[7] Moschakis I.A., Karatza H.D., Evaluation of gang scheduling performance and cost in a cloud computing system, The Journal of Supercomputing (2012), 975-992.

[8] Zhang F., Cao J., Li K., Khan S.U., Hwang K., Multi-objective scheduling of many tasks in cloud platforms, Future Generation Computer Systems (2014), 309-320.

[9] Achar R., Thilagam P., Shwetha D., Pooja H., Optimal scheduling of computational task in cloud using Virtual Machine Tree, Third International Conference on Emerging Applications of Information Technology (EAIT) (2012), 143-146.

[10] Anglano C., Canonico M., Scheduling algorithms for multiple bag-of-task applications on desktop grids: A knowledge-free approach, IEEE International Symposium on Parallel and Distributed Processing (2008), 1-8.

[11] Sulistio A., Buyya R., A time optimization algorithm for scheduling bag-of-task applications in auction-based proportional share systems, 17th International Symposium on Computer Architecture and High Performance Computing (2005), 235-242.

[12] Wieczorek M., Podlipnig S., Prodan R., Fahringer T., Bi-criteria scheduling of scientific workflows for the grid, 8th IEEE International Symposium on Cluster Computing and the Grid (2008), 9-16.

[13] Lavc M., Strategies for dynamic workflow scheduling on grids, PhD, COPPE/UFRJ, COPPE/UFRJ, (2007).

[14] Ramakrishnan L., Reed D.A., Performability modeling for scheduling and fault tolerance strategies for scientific workflows, Proceedings of the 17th international symposium on High performance distributed computing (2008), 23-34.

[15] Yu J., Buyya R., Tham C.K., Cost-based scheduling of scientific workflow applications on utility grids, First International Conference on e-Science and Grid Computing (2005), 1-9

[16] Xu M., Cui L., Wang H., Bi Y., A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing, IEEE International Symposium on Parallel and Distributed Processing with Applications (2009), 629-634.

[17] Bandini M., Mury A.R., Schulze B., Salles R., A Grid QoS Decision Support System Using Service Level Agreements, In Congresso da SociedadeBrasileira de Computacao de, Sociedade Brasileira de Computacao: Porto Alegre, RS, Brazil, (2009).

[18] Lin H.C., Raghavendra C., An approximate analysis of the join the shortest queue (JSQ) policy, IEEE Transactions on Parallel and Distributed Systems (1996), 301-307.

[19] Dean J., Ghemawat S., MapReduce: a flexible data processing tool, Communications of the ACM (2010), 72-77.

[20] Casas I., Taheri J., Ranjan R., Wang L., Zomaya A.Y., A balanced scheduler with data reuse and replication for scientific workflows in cloud computing systems, Future Generation Computer Systems, (2016).

[21] Tsakalozos K., Kllapi H., Sitaridi E., Roussopoulos M., Paparas D., Delis A., Flexible use of cloud resources through profit maximization and price discrimination, IEEE 27th International Conference on, Data Engineering (ICDE) (2011), 75-86.

[22] De Oliveira K.A., Ocaña, F.B., Mattoso M., A provenance-based adaptive scheduling heuristic for parallel scientific workflows in clouds, Journal of grid Computing (2012), 521- 552.

[23] Topcuoglu H., Hariri S., Wu M.Y., Performance-effective and low-complexity task scheduling for heterogeneous computing, IEEE Transactions on Parallel and Distributed Systems (2002), 260-274.

[24] Syed Navaz A.S., Jayalakshmi P., Asha N., Optimization of Real-Time Video Over 3G Wireless Networks, International Journal of Applied Engineering Research 10 (18) (2015), 39724 – 39730.

[25] Syed Fiaz A.S., Asha N., Sumathi D., Syed Navaz A.S., Data Visualization: Enhancing Big Data More Adaptable and Valuable, International Journal of Applied Engineering Research 11 (4) (2016), 2801-2804.

[26] Syed Navaz A.S., Dr. Kadhar Nawaz G.M., Flow Based Layer Selection Algorithm for Data Collection in Tree Structure Wireless Sensor Networks, International Journal of Applied Engineering Research 11 (5) (2016), 3359-3363.

[27] Syed Navaz A.S., Dr. Kadhar Nawaz G.M., Layer Orient Time Domain Density Estimation Technique Based Channel Assignment in Tree Structure Wireless Sensor Networks for Fast Data Collection, International Journal of Engineering and Technology 8 (3) (2016), 1506-1512.

[28] Syed Navaz A.S., Asha N., Sumathi D., Energy Efficient Consumption for Quality Based Sleep Scheduling in Wireless Sensor Networks, ARPN Journal of Engineering and Applied Sciences 12 (5) (2017), 1494-1498.

[29] Anusha Priya A., Gunasundari R., Securing Data on the Cloud Server by the User Authentication and Data Security Techniques, International Journal of Computer Applications 165 (4) (2017), 6-12.

[30] Yasmin S., Anusha Priya A., Decentralized Entrance power with Secret Endorsement of data Stored in Clouds, International Journal of Innovative research in Computer and Communication Engineering 3 (8) (2015), 7279-7284.

[31] Yamuna V., Anusha Priya A., Efficient and Secure Data Storage in Cloud Computing RSA and DSE Function, International Journal of Innovative Research in Computer and Communication Engineering 3 (7) (2015), 6758-6763.

[32] Anusha Priya A., Mohanapriya A., An Effective Scrutiny of Static and Dynamic Load Balancing In Cloud, International Journal of Emerging Technology in Computer Science & Electronics 23 (4) (2016), 27 -30.

[33] Usha M., Akilandeswari J., Syed Fiaz A.S., An efficient QoS framework for Cloud Brokerage Services, IEEE International Symposium on Cloud and Service Computing, (2012), 76-79.

[34] Anusha Priya A., Gunasundari R., Ensemble Secure Communication Protocol (ESCP) for Data Storage and Retrieval in Private Cloud, International Journal of Applied Engineering Research 12 (21) (2017), 11294-11301.

[35] Anusha Priya A., Gunasundari R., Advanced encryption standard algorithm based distributed data storage security mechanism (D2S2M) for private cloud, ARPN Journal of Engineering and Applied Sciences 12 (20) (2017), 5708-5715.