IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

ITRP: Instant Time Resource Provisioning Prototype for Adequate Quality of Service

Mrs. Shilpa[1], Mrs. Chetana Srinivas[2]

M.Tech Student, Department of CSE, East West Institute of Technology, Bangalore, India[1]

Assistant Professor, Department of CSE, East West Institute of Technology, Bangalore, India[2]

Abstract: In the era of Cloud computing where it utilizes Infrastructure as a Service (IaaS), elasticity has become one of the very essential components for providing better Quality of Services (QoS). The concept of elasticity ensures an effective and dynamic resource allocation during sudden changes of workloads in Virtual Machines (VMs). The current research trends highlight that there are very less efficient virtualized environments for proper task scheduling in the field of cloud computing. Therefore effective resource management/provisioning during overload of jobs in the distributed virtual machines have become one of the most challenging tasks. It can be seen that most of the existing techniques cannot respond rapidly when the work load associated with a particular service amplifies. Most of the exiting trends are found to have inaccuracy in case of taking proper decisions which results problems during resources provisioning in cloud services. In this study an efficient Instant Time Resource Provisioning Prototype (ITRP) has been introduced which increases the scalability of allocating resources during each alteration cycle when work load increases. The performance analysis of the proposed system shows that it achieves very accuracy in achieving speed up for resource provisioning in cloud infrastructures as compare to the existing models.

Keywords: Resource Provisionong , Cloud Computing , Virtual Machines.

I. INTRODUCTION

Cloud computing has considered as an infrastructure of modern collaborative computing services and resource provisioning configured with internet for allowing individuals and business to utilize the trusted third party components in a distributive manner. Therefore users from remote locations also will be able to access the shared hardware and software components managed by various third parties in cloud infrastructures [1]. The Cloud computing model allows reliable access to the useful information from anywhere in the world and it also provides a pool of resources such as online file storage, social networking platforms, email and shared e-drives etc. Some of the existing studies derive cloud computing infrastructure as on demand network services [2]. Cloud computing enables the dynamic resource sharing services in an virtualized environment , it also ease the on demand access of pool of configurable computing resources such as e.g. storage applications and various services in order to achieve minimal management a effort with respect to efficient services [3].

In the era of cloud computing various resources and services are collaborated in a virtualized manner where the virtualized network has been integrated with physical location of each and every virtual machines, size and implementation associated with each resources [4][5]. However cloud computing has been conceptualized in order to provide extensibility , dynamicity and elasticity base real time resource provisioning but elastic resource provisioning has become one of the most challenging tasks in the field modern infrastructure as a service (IaaS) for cloud environment.

The following figure highlights that how data centers and virtual machines can be utilized efficiently in order to achieve better throughput and response.

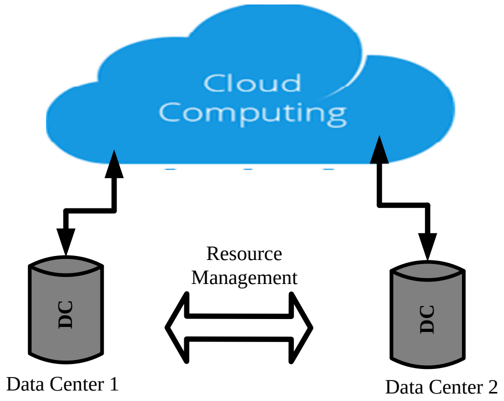

Figure 1 Resource Management in Cloud

It can be seen that workloads hosted by various services in the virtual machines sometimes degrade the performance of the overall systems which results inefficient quality of services with respect to the delay in response time. The imbalanced big data management system suffers a lot during the pipelined execution of workloads in dynamic runtime while job arrivals to VM and the resources are not configured as per the respective requirements [6]. For mitigating these issues this study aims to propose an instant resource provisioning prototype in order to achieve an efficient resources provisioning and QoS in cloud computing. The experimental outcomes highlight the effectiveness of the proposed model. The paper is

DOI 10.17148/IJARCCE.2016.5525

organized as follows Section II discusses about the recent studies towards existing load balancing techniques towards service level agreement of cloud computing which is followed by problem statement in Section III. Section IV discusses about research methodology followed by discussion of algorithm implementation in Section V. Section VI discusses about the result analysis followed by conclusion in Section VII.

II. RELATED WORK

This section discusses about the existing studies that has been carried out in the past regarding achieving efficient work load balancing and resource provisioning in cloud computing. The discussion in this section is carried out with respect to recent work being carried out for aggressive resource provisioning in virtualized environments.

The study of **Garg et al [7]** proposed a framework and a procedure which measures the quality and organizes the cloud administration. Creators propose that the structure will make a critical effect furthermore make a sound rivalry among cloud administration suppliers keeping in mind the end goal to fulfill Service Level Agreement (SLA) and improves the QoS. Through contextual analyses the creators have demonstrated relevance of the positioning system.

Ardagna et al [8] performed a study on QoS demonstrating approaches; creators additionally evaluated and arranged earlier application to few basic leadership issues emerging in cloud QoS administration.

Apkan and Vadhanam [9] have exhibited a study on the nature of administration in distributed computing as for strategies utilized, points of interest and burdens.

Irugurala and Chatrapati [10] introduced booking calculations for productive asset designation to augment benefit and client level fulfillment for SaaS suppliers. Through reproduction, creators demonstrated that the calculations function admirably in various situations. By reproduction creators have appeared in normal the Prof PD calculation gives greatest benefit (spares around 40% of VM expense) in contrasted with all proposed calculations by modifying all sort of QoS parameters.

Sharkh et al [11] examined variables that influence the execution of asset assignment models. Creators have talked about these elements in subtle element, and pointed out examination crevices. Outline difficulties are talked about with the point of giving a reference to be utilized when planning an exhaustive vitality mindful asset portion model for distributed computing server farms.

Gouda et al [12] proposed another methodology that allots asset with least wastage and gives most extreme benefit. The created asset allotment calculation depends on various parameters like time, cost, No of processor solicitation and so on. The created need calculation is utilized for a superior asset allotment of employments in the cloud environment utilized for the reenactment of various models or occupations in an effective way. After the effective asset designation different occupations creators have additionally assessed execution. Creators performed execution investigation of the considerable number of calculations in different frameworks and contextual investigations.

Jangra and Saini [13] proposed planning calculation which measures both asset expense and calculation execution and enhances the calculation and correspondence proportion by the client undertakings as per a specific cloud asset's preparing ability and sends the errands occupations to the assets.

Vakilinia et al [14] acquainted a calculation with dispense remote interfaces and cloud. The proposed model depends on the Wireless Network Cloud (WNC) idea. Creators recommend that the proposed model considers power utilization, application nature of administration (QoS) profiles, and comparing cost capacities. Creators have utilized a multi-target improvement approach utilizing an occasion based limited state model and element requirement programming technique to decide suitable transmission power, process power, cloud offloading and ideal QoS profiles. Numerical results demonstrate that the proposed calculation spares the versatile battery life and insurances both QoS and cost at the same time. Besides, it decides the best accessible cloud server assets and remote interfaces for applications in the meantime.

Shyamala and Rani [15] uncovered how viably asset designation issue can be tended to in the point of view of cloud administration supplier furthermore gave a similar examination which causes in selecting parameters to meet the target capacity for enhancing the interest to augment the benefit.

Katyal and Mishra [17] presented specific calculation for assignment of cloud assets to end-clients on-interest premise. This calculation depends on min-min and max-min calculations. Creators have utilized two customary errand booking calculation. The particular calculation utilizes certain heuristics to choose between the two calculations with the goal that general make range of undertakings on machines is minimized. The assignments are planned on machines in either space shared or time shared way. Creators assess their provisioning heuristics utilizing a cloud test system, called CloudSim. Creators additionally contrasted our methodology with the insights got when provisioning of assets was done in First-Cum-First-Serve (FCFS) way. The trial results demonstrate that general make range of undertakings on given arrangement of VMs minimizes altogether in various situations.

Shu et al [18] proposed an enhanced clonal choice calculation in light of time expense and vitality utilization models in distributed computing environment. Creators have investigated the execution of their methodology utilizing the CloudSim toolbox. The test results demonstrate that their methodology has massive potential as it offers critical change in the parts of reaction time and make range, exhibits high potential for the change in vitality effectiveness of the server farm, and can viably meet the administration level understanding asked for by the clients.

DOI 10.17148/IJARCCE.2016.5525

III. PROBLEM STATEMENT

The previous section introduced various existing studies towards work load balancing in virtualized and cloud environment. The problem considered is to design and develop a method of resource allocation where, the method adopts a mechanism of adjusting Virtual Machine (VM) instances dynamically by speeding up resource provision by using aggressive strategy where the resource are over-provisioned to fulfill the required QoS and then speed down the resource provision if required. The quality of services in the cloud environment suffers a lot due to lack of efficient resource provisioning and scalable elasticity in virtualized environment. It is necessary to design a cost effective as well as efficient decision provider in assisting the resource management to improve elasticity. The issues associated with resource provisioning in cloud computing are stated below.

1. The resource allocation in par with the workload.
2. Improvement of the decision adaptability time should be evaluated.
3. Enhancement of the efficiency and performance of the VM's.
4. QoS in virtualized environment.

Issues due to Elasticity:
Resources provisioning time
One potential issue is that flexibility requires some investment. A cloud virtual machine (VM) can be obtained whenever by the client, in any case, it might take up to a few minutes for the procured VM to be prepared to utilize. The VM startup time is reliant on variables, for example, picture size, VM sort, server farm area, number of VMs, etc. Cloud suppliers have distinctive VM startup execution. This infers any control system intended for flexible applications must consider in its choice process the time required for the versatility moves to make impact, for example, provisioning another VM for a particular application segment [19].

Monitoring elastic applications:
Versatile applications can assign and de allots assets, (for example, VMs) on interest for particular application segments. This makes cloud assets unstable, and conventional checking apparatuses which relate observing information with a specific asset (i.e. VM, for example, Ganglia or Nagios, are no more appropriate for checking the conduct of versatile applications. For instance, amid its lifetime, an information stockpiling level of a flexible application may include and expel information stockpiling VMs because of expense and execution prerequisites, shifting the quantity of utilized VMs. Accordingly, extra data is required in checking flexible applications, for example, partner the intelligent application structure over the basic virtual framework. This thusly produces different issues, for example, how to total information from various VMs towards removing the conduct of the application part running on top of those VMs, as various measurements may should be amassed in an unexpected way (e.g., cpu use could be arrived at the midpoint of, system exchange may be summed up) [20].

Need of Elasticity:
While sending applications in cloud bases (IaaS/PaaS), prerequisites of the partner should be considered keeping in mind the end goal to guarantee legitimate versatility conduct.

Despite the fact that generally one would attempt to locate the ideal exchange off amongst expense and quality or execution, for true cloud clients necessities with respect to the conduct are more intricate and focus on various measurements of versatility.

Multiple levels of control:
Cloud applications can be of shifting sorts and complexities, with various levels of relics sent in layers. Controlling such structures must mull over an assortment of issues, a methodology in this sense being For multi-level control, control frameworks need to consider the effect lower level control has upon more elevated amount ones and the other way around (e.g., controlling virtual machines, web holders, or web administrations in the same time), and also clashes which may show up between different control methodologies from different levels [21].

IV. RESEARCH METHODLOGY

The proposed study aims to develop a novel and scalable instant time resource provisioning mechanism in order to achieve efficient and elastic resource provisioning in the field of cloud computing environment.
The design specification associated with the proposed system is highlighted in the figure 2 which shows that the proposed system includes the following components.

1. Physical machine: Physical machines are considered as hardware component which have a physical existence and configured with necessary components of virtual machines for performing computation.

2. Hypervisor: The proposed study also uses a hypervisor module which usually computer software used for monitoring the work status associated with each VMs. Each and every VM are well managed with a guest operating system will executes on an virtualized platform in order to create multiple instances for sharing virtualized hardware resources very efficiently.

3. Virtual Machine (Vm) : Virtual Machines are basically real time computers operated on the concept of real or hypothetical computer and their respective implementation. Basically this module is composed of different components which involve specialized hardware, software and the combination of both.

4. Decision Engine: This module basically executes a computer algorithm which generates a decision that how efficiently resource allocation in Virtualized environments can be reconfigured. This module basically import the knowledge based data from the memory table and work performance metric which notify it about current work load status of each and every VMs.

DOI 10.17148/IJARCCE.2016.5525

5. Scheduler: a scheduler is also a computer program which basically maintain the status of work allocated by virtual computation elements e.g. treads , processes , data flows , it also keep tracking of various other hardware resources such as processors and cloud based network links.

6. VM Controller: This model has been executed to manage VMs with respect to work load and resource provisioning. It provide the live performance and resource utilization status where it usually track down the work performance and load metrics from scheduler , service load status decision engine and memory table.

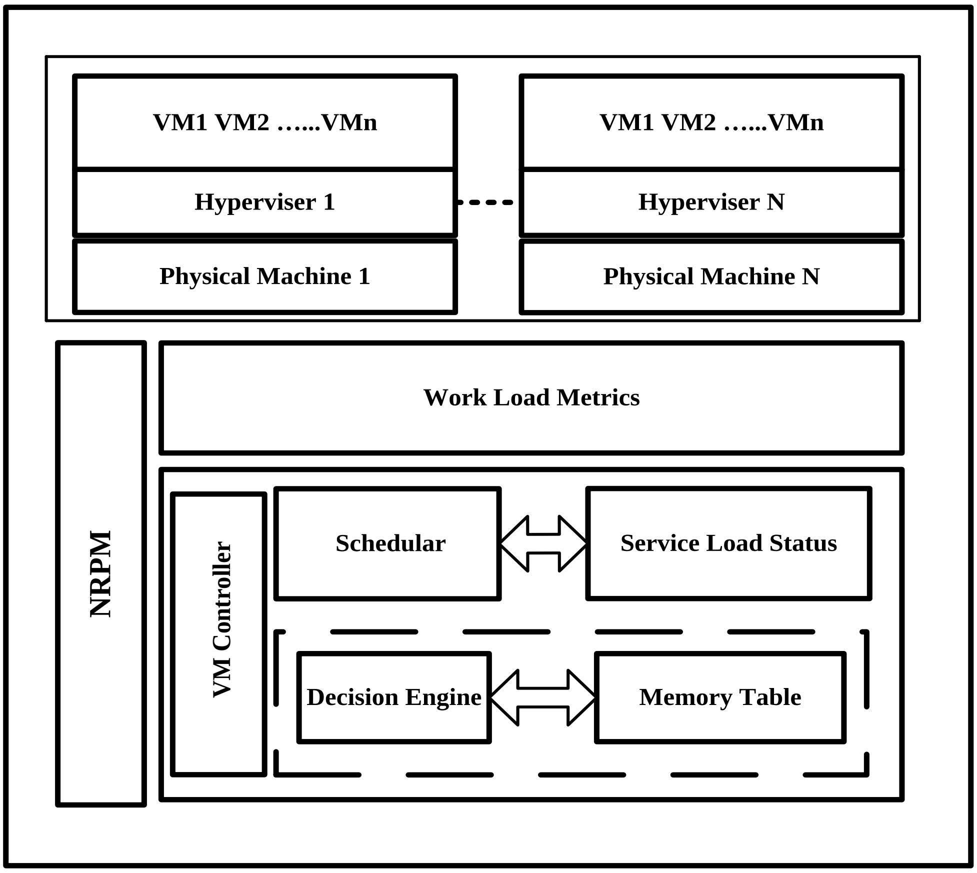

Figure 2 Schematic diagram of the proposed system

Above stated figure shows a tentative architecture of the proposed system which shows that the proposed system has been conceptualized in a way thus it can work along with the individual cloud services deployed in a network. The proposed system allocates VMs to the users by evaluating different type of requirements such it continuously track down the service status requests and jobs allocated to each VM. It also can improve the cloud service throughput efficiency by managing the main components which are physical machines, work load metrics and the VM controller modules. The proposed system basically employs an alternate service for collecting the work load metrics from each individual virtual machine. The alternate service usually collects the information associated with the CPU and memory utilization metrics and then it sends the respective data to the VM controller and the scheduler.

The VM controller and scheduler compute the service load status. The current service load status maintains a table where it indicates the previous load status in each VM. The proposed system also maintains another module which is termed as decision making model; it uses the current service load statues for managing and reconfiguring the resource allocations. The effectiveness of the decision has been judged based on a reward strategy. The workload metrics and respective action strategies are stored and maintained in a table called CMAC which are further utilized in an instant time based aggressive resource allocation strategy. The scheduler usually updates the VM controller about the work load status and based on that status and previous information stored in memory table the proposed system efficiently allocates the resources.

V. IMPLEMENTATION

The implementation of the proposed system has been carried out using Netbeans tool and Java programming language. The implementation of the proposed system has been considered using physical machines and virtual machines where each and every physical machine consists of 4 CPUs.

The implementation of the proposed system in a real time prototype shows that how work load metrics has been evaluated considering memory utilization by each CPU, processor speed associated with each physical machine and their respective virtual machines (VMs).

DOI 10.17148/IJARCCE.2016.5525

The pseudo codes of each and every module for implementing the proposed system are given below.

Pseudo codes of class physical hardware units

```
Start
Initialize class PhysicalHwUnits
Define ← physical hardware  id;
Define ← physical hardware memory = 32768 MB;
Define ← physical hardware storage = 204800 MB
Define ←bandwidth = 10000;
Define ← processor speed in jobs/sec
Initialize no of CPUs in each physical hardware = 4;
Evaluate the method physicalhwunit ()
Apply get ()→hardware memory
Apply set ()→hardware memory
Apply get ()→hardware id
Apply set ()→hardware id
Apply get ()→ processor speed;
Apply set ()→processor speed;
Apply get ()→ No of CPUs
Apply set ()→ No of CPUs;
End
```

The above pseudo code highlights that how physical hardware in a cloud computing services are configured using various physical attributes.

Pseudo code for implementing Virtual Machine

```
Start
Define  ← VM characteristics
Define ← Host;
Get() ← $VM_{id}$
Return $VM_{id}$
Gets the number of CPUs required by the VM
return number of CPUs
Sets the number of CPUS required by this VM
Define ← No of CPUs
Gets the amount of memory used by this VM
return amount of memory used by the VM
Gets the amount of storage used by this VM
return amount of storage used by the VM
Gets the amount of bandwidth used by this VM
return amount of bandwidth used by the VM
Set the bandwidth used by a VM
Sets the physical machine  that runs this VM
Host running the VM
Returns an object of the type VMCharacteristics,
Returns a reference to the scheduler in use by the VM
End
```

The above stated pseudo code highlights how virtual machine can be implemented in a cloud platform. It also shows that

Pseudo code for VM Controller Module

```
Start
Import ←cloud.core.pm
Import ← util.package
Initialize class VM Controller
Initialize current job = 0;
Initialize new job = 0;
Define ← Hash_table
Define ← $Min_{job}$
Define ← $Max_{job}$
Evaluate ← VM Controller
Check (Analysis Result)
If  ($VM_n$ ← idle)
Evaluate VM characteristics
Allocate job;
Aggressive resource allocation
End
```

VI. RESULT DISCUSSION

This section discusses about the important findings of the proposed study. It also highlights the experimental outcomes of the proposed ITRP model. The following table 1 show how efficiently resources (CPUs, VMs, processors) have been utilized and allocated considering random job arrivals. The proposed system considers 4 CPUs in each physical machine and it also shows aggressive resource provisioning during the maximum and minimum job arrivals.

Table-I

```
DC1_hw_0=> Memory :32768 MB, Storage :2048000 MB
, Processor Speed :10000, CPU :4
 DC1_hw_1=> Memory: 32768 MB, Storage :2048000
MB ,Processor Speed :10000, CPU :4 -----------
 Data Center DC1 : Physical H/w 2 Total memory :65536
MB, Storage :4096000 MB , Bw :2000000 , CPU :8
 Service load Status :  , CPU : 179.46 % , Memory
:8973000.0
  Allocated CUP : 89.72
  Allocated CUP : 89.72
 Service load Status :  , CPU : 106.82000000000001 % ,
Memory :5341000.0
 Allocated CUP : 53.4
Allocated CUP : 53.4
 Service load Status :  , CPU : 114.06 % , Memory
:5703000.0
Allocated CUP : 57.02
 Allocated CUP : 57.02
Service load Status :  , CPU : 201.96 % , Memory
:1.0098E7
Allocated CUP : 100.98
Allocated CUP : 100.98
 Service load Status :  , CPU : 60.800000000000004 % ,
Memory :3040000.0
Allocated CUP : 60.800000000000004
Allocated CUP : 60.800000000000004

Service load Status :  , CPU : 33.7 % , Memory :1685000.0
 Allocated CUP : 33.7
Allocated CUP : 33.7

Service load Status :  , CPU : 160.96 % , Memory
:8048000.0
 Allocated CUP : 80.48
Allocated CUP : 80.48
```

The following figure 3 shows the comparative analysis of the proposed system which has been performed considering the existing SPRNT aggressive resource provisioning model. The comparative analysis highlights the effectiveness of the proposed system with respect to CPU utilization.

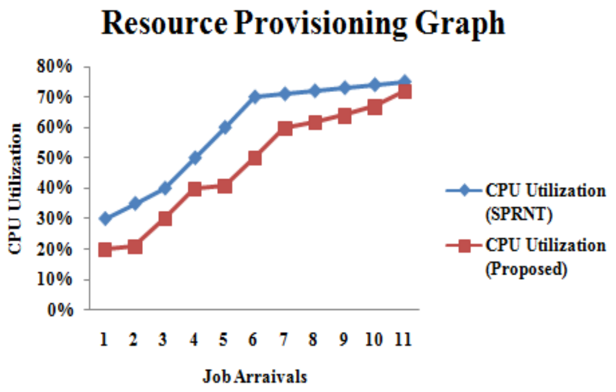

Figure 3 comparative analysis

The proposed ITRP model utilizes very less amount of CPU compare to existing SPRNT for the same amount of job arrival.

VII. CONCLUSION

In the recent time efficient resource provisioning in cloud computing has become very challenging task. This paper propose an instant time resource provisioning prototype in order to utilize the Virtual Machines in cloud infrastructure. The experimental outcomes of the proposed system have been highlighted in the above mentioned section VI. The comparative analysis shows that the proposed system works much efficiently as compare to the existing SPRNT model.

REFERENCES

[1] R. N. Calheiros, R. Ranjan, and R. Buyya, “Virtual machine provisioning based on analytical performance and QoS in cloud computing environments,” in Proc. Int. Conf. Parallel Process., 2011 pp. 295–304.

[2] Q. Zhang, M. Zhani, R. Boutaba, and J. Hellerstein, “Dynamic heterogeneity- aware resource provisioning in the cloud,” IEEE Trans. Cloud Comput., vol. 2, no. 1, pp. 14–28, Mar. 2014.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A view of cloud computing,” Commun. ACM, vol. 53, no. 4, pp. 50–58, 2010.

[4] H. Liu, Y. Zhang, Y. Zhou, X. Fu, and L. T. Yang, “Receiving buffer adaptation for high-speed data transfer,” IEEE Trans. Comput., vol. 62, no. 1, pp. 2278–2291, Nov. 2013.

[5] H. Nguyen, Z. Shen, X. Gu, S. Subbiah, and J. Wilkes, “AGILE: Elastic distributed resource scaling for infrastructure-as-aservice,” in Proc. 10th Int. Conf. Autonom. Comput., 2013, pp. 69–82.

[6] N. R. Herbst , S. Kounev, and R. Reussner, “Elasticity in cloud computing: What it is, and what it is not,” presented at the Proc. 10th Int. Conf. Auton. Comput. San Jose, CA, USA, 2013.

[7] Garg, S.K., Versteeg, S. and Buyya, R., 2011, December. SMICloud: a framework for comparing and ranking cloud services. In Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on (pp. 210-218). IEEE.

[8] Abdelmaboud A, Jawawi DN, Ghani I, Elsafi A, Kitchenham B. Quality of service approaches in cloud computing: A systematic mapping study. Journal of Systems and Software. 2015 Mar 31;101:159-79.

[9] Akpan, H.A. and Vadhanam, B.R., A Survey on Quality of Service in Cloud Computing. International Journal of Computer Trends and Technology (IJCTT) volume, 27, pp.58-63.

[10] Lee, Y.C., Wang, C., Zomaya, A.Y. and Zhou, B.B., 2012. Profit-driven scheduling for cloud services with data access awareness. Journal of Parallel and Distributed Computing, 72(4), pp.591-602.

[11] Irugurala, S. and Chatrapati, K.S., 2013. Various Scheduling Algorithms for Resource Allocation In Cloud Computing. The International Journal Of Engineering And Science (IJES), 2, pp.16-24.

[12] Abu-Sharkh, S. and Doerffel, D., 2004. Rapid test and non-linear model characterisation of solid-state lithium-ion batteries. Journal of Power Sources, 130(1), pp.266-274.

[13] Wong, C.K., Gouda, M. and Lam, S.S., 2000. Secure group communications using key graphs. Networking, IEEE/ACM Transactions on, 8(1), pp.16-30.

[14] Saini, N., Yadav, J.S., Jangra, S.K., Sharma, D. and Sharma, V.K., 2011. Thermodynamic studies of molecular interactions in mixtures of o-toulidine with pyridine and picolines: Excess molar volumes, excess molar enthalpies, and excess isentropic compressibilities. The Journal of Chemical Thermo dynamics, 43(5), pp.782-795.

[15] Chen, T.Y., Vakilinia, K., Divsalar, D. and Wesel, R.D., 2015. Protograph-based raptor-like LDPC codes. Communications, IEEE Transactions on,63(5), pp.1522-1532.

[16] Harshavardhan, D., Rani, T.S., Ulaganathan, K. and SEETHARAMA1, N., 2002. An Improved Protocol for Regeneration of Sorghum bicolor from Isolated Shoot Apices. Plant Biotechnology, 19(3), pp.163-171.

[17] Katyal, M. and Mishra, A., 2014. A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment. arXiv preprint arXiv:1403.6918.

[18] Lin, Shu, and Daniel J. Costello. Error control coding. Pearson Education India, 2004.

[19] N. R. Herbst , S. Kounev, and R. Reussner, “Elasticity in cloud computing: What it is, and what it is not,” presented at the Proc. 10th Int. Conf. Auton. Comput., San Jose, CA, USA, 2013.

[20] P. Bodik, A. Fox, M. J. Franklin, M. I. Jordan, and D. A. Patterson, “Characterizing, modeling, and generating workload spikes for stateful services,” in Proc. 1st ACM Symp. Cloud Comput., 2010, pp. 241–252.

[21] E. Casalicchio, D. A. Menasce´, and A. Aldhalaan, “Autonomic resource provisioning in cloud systems with availability goals,” inProc. Int. Conf. Cloud Autonom. Comput., 2013, p. 1.

DOI 10.17148/IJARCCE.2016.5525