Volume 6, Issue 9, September 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Autonomic Resource Provisioning Algorithm for Cloud Computing using Match Making Technique

G. Sobers Smiles David
Asst.Professor and Head, PG Dept. of Computer Science,
Bishop Heber College (Autonomous)
Trichirappalli, TN, India

Dr. R. Anbuselvi
Asst. Professor, PG Dept. of Computer Science,
Bishop Heber College(Autonomous)
Trichirappalli, TN, India

Abstract— The last decade has witnessed a huge number of innovations and new technologies which have impacted our routines and have simplified our lives. The emergence of cloud computing has resulted in a paradigm shift the way data is stored and handled. The scale and magnitude of cloud computing is vast, and the process of managing the Cloud, be it resource utilization or system management is too complex. In order to provide highly reliable and secure services, automated and intelligent mechanisms are needed for managing the resources in the cloud. This paper explores the issues and challenges in autonomic resource provisioning and management and proposes an enhanced algorithm using match making technique.

Keywords— Autonomic Cloud Computing, Resource Management, Resource Scheduling.

I. INTRODUCTION

In 1969, A vision for delivering “computing as a utility” was introduced by Leonard Kleinrock, the chief scientist of the original Advanced Research Project Agency (ARPA) project. Kleinrock envisioned that computer networks would be used as a “utility” [1]. The birth of the Internet occurred in Prof.Leonard Kleinrock UCLA laboratory(3420 Boelter Hall) when his Host machine became the first computer and first node of the Internet in September 1969. From 1969, Information and Communication Technology (ICT) has made many advances in various areas to make this vision a reality [2]. The advances in networked computing environments have transformed computing to a model consisting of services that can be commoditized and delivered similarly to utilities such as water, electricity, gas, and telephony [3].

There was no looking back since then. Internet has grown manifold and the emergence of mobile technology has made internet access available to 80% of human population. Today, Internet has evolved into Cloud Computing, offering computing capabilities on the go. Cloud computing has revolutionized every domain in the business. The National Institute of Standards and Technology(NIST) of the U.S government defines Cloud Computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable resources(eg., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. NIST’s cloud model consists of five essential characteristics, three service models and four deployment models. The five characteristics are on-demand self- service, broad network access, resource pooling, rapid elasticity, and measured service. The three service models are Cloud Software as a Service (SaaS), Cloud Platform as a Service (PaaS), and Cloud Infrastructure as a Service (IaaS). The four deployment models are Private Cloud, Public Cloud, Community Cloud and Hybrid Cloud.

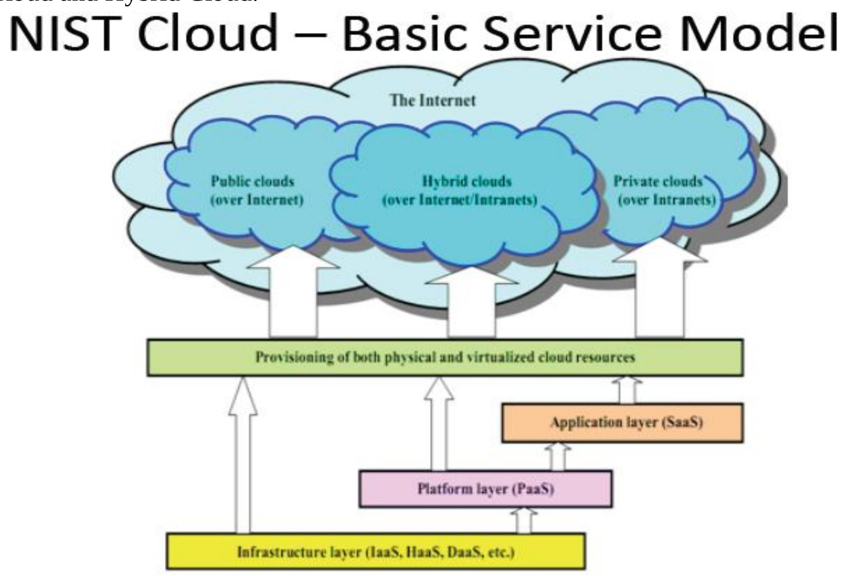
NIST Cloud – Basic Service Model
The Internet
Public clouds
(over Internet)
Hybrid clouds
(over Internet/Intranets)
Private clouds
(over Intranets)
Provisioning of both physical and virtualized cloud resources
Application layer (SaaS)
Platform layer (PaaS)
Infrastructure layer (IaaS, HaaS, DaaS, etc.)

Fig 1.NIST Cloud-Basic Service Model

Cloud Computing is presumed to offer computing as an utility [4] in the next generation of IT platforms. Apart from IaaS, PaaS, and SaaS it also provides XaaS. XaaS is anything as a service. This includes AaaS (Architecture as a Service), DaaS (Data as a Service), NaaS (Network as a Service), HaaS (Hardware as a Service), RaaS (Recovery as a Service), Vaas (Voice as a Service), FaaS (Framework as a Service), and the list is endless [5]. Thus, Cloud is a combination of the above business developments and technologies such as Virtualization, Grid Computing, Web Services and utility computing. Cloud Services could be outsourced to Cloud Service Providers deployment and maintenance of specialized cloud products and services. The main advantage would be to pay only for the services that the user has availed. Cloud Computing has revolutionized the way data is stored and accessed. It also promises the capability to offer quality computing services which could be subscribed. As the number of users is increasing exponentially, Resource Management Systems (RMS) play a vital role in ensuring that the Service Level Agreements with clear QoS (Quality of Service) parameters are enforced. The RMS balances the service requests from the users and the expected service performance from the provider. System oriented approaches in resource allocation maximize system usage and job performance but fail to deliver on-demand-service computing. Before cloud computing,

ICT administrator’s job was easy, as the only objective of resource provisioning was the performance. The complexity involved in resource provisioning has grown exponentially manifold due to increased complexity [6]. The Complexity mainly arises from the heterogeneity of the users and their requests. The traditional resource management model is not capable of processing the task of resource assignment and allocating resources dynamically [7]. As Cloud offers the capability to access information anytime, anywhere and anyhow, it is difficult for a cloud service provider to dynamically allocate resources efficiently. In order to provide highly reliable and secure services, automated and intelligent mechanisms are needed for managing the resources in the cloud.

II. RESOURCE MANAGEMENT IN CLOUD COMPUTING

A. Background

In Cloud Computing, Resource management involves Resource Provisioning, Resource Allocation and Resource Monitoring. Cloud resources include the web servers, memory, storage, network, CPU, application servers, and virtual machines. Virtual Machines are the processing units in Cloud. Virtualization offers scope for solutions to manage resources but increases complexity [8].

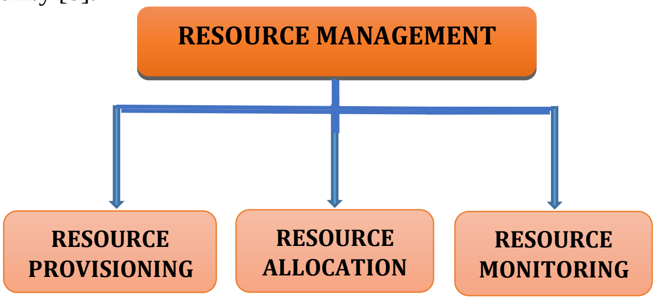

Fig 2.Resource Management Stages in Cloud Computing

Resource Provisioning is the process of resource discovery. Resource Allocation involves the process of allocating resources. It is done in two ways, Static and Dynamic. In Resource monitoring the allocated resources are monitored. When a service request is accomplished successfully, the resource is released and given to the resource provisioning module.

The resources in a cloud could be categorized into two as Open and Vague. Open Systems are the ones whose local resource scheduling policy is known openly, it is not so in Vague [8].

B. Autonomic Cloud Computing

The performance of any system depends on the effective management of resources. This is particularly significant in cloud computing systems which involves management of large number of Virtual machines and Physical machines. In particular, the performance is inherently dependant on effective provisioning of resources [9]. Significant performance degradation is caused by resource contention by multiple applications. The heterogeneous nature of hardware resource in the cloud make it even more challenging. Any resource management system in cloud computing must cater to the three service types IaaS, PaaS, and SaaS. For IaaS, the Resource Management System(RMS) should be adaptive for ever changing virtual resource requirements and resource restrictions. Also, the RMS should be capable of handling complex multiple resource to performance relationship without any assumption of any system model. The RMS system must be highly scalable to support large scale applications. For PaaS, the RMS should provide optimized performance of the platforms in offer. It should also provide mitigation for virtual resource contention and job interference. For SaaS, the RMS should offer flexibility for varying application demands for resources, and be highly adaptive and provide online configuration operations [10].

Autonomic Cloud systems are self-managing systems with capabilities for self-regulating, self-healing, self-protecting, and self-improving machanisms. Because of the scale and heterogeneity of cloud resources, Autonomic Computing offers excellent options for any autonomic model developed. In the proposed Autonomic Resource Provisioning model, it is planned to provide an autonomic provisioning model and subsequent management techniques.

III. AUTONOMIC CLOUD COMPUTING ARCHITECTURE

A. Previous Work

Autonomic management [2], [6] is the most desired feature in any distributed computing environment such as clouds. Autonomic systems are self managing systems with abilities such as self-healing, self-improving, self-regulating, and self-protecting. Preliminary investigation in the development of autonomic resource provisioning model is already made by Academia and Industry. Parashar and Harini [5], provided an overview of early autonomic systems in storage management(OceanStore[7],

Storage Tank[8]), Computing resources(Oceano[18]), and DBMS (Smart DB2[10]). One of the main beneficiary of Autonomic systems is Computing grids, which has benefited the maximum]12]. CometCloud[11] provided an infrastructure model for automatic management of workflow applications in Cloud. Other works [13],[14],[15] explored in depth, the resource provisioning in cloud applications. However, these works do not offer an integrated environment for cost effective autonomic resource provisioning with efficient security features. Amazon’s Elastic MapReduce provides options for its customers to add or remove nodes depending upon the changing needs of their job flow. But, this service does not provide autonomic provisioning.

B. Factors to be considered

While developing an Autonomic Cloud Computing Architecture, the following things need to be taken care of [11].

- To develop an Autonomic Resource management system and supporting algorithms that support dynamic resource provisioning with maximum efficiency and minimum cost.
- To design mechanisms to ensure that the legitimate users are allocated resources.
- To develop co-ordination mechanisms for resource provisioning to ensure that the user’s Quality of Service (QoS) parameters are satisfied without compromising on cost. The Architecture of Autonomic Cloud Management System is shown in Fig.4

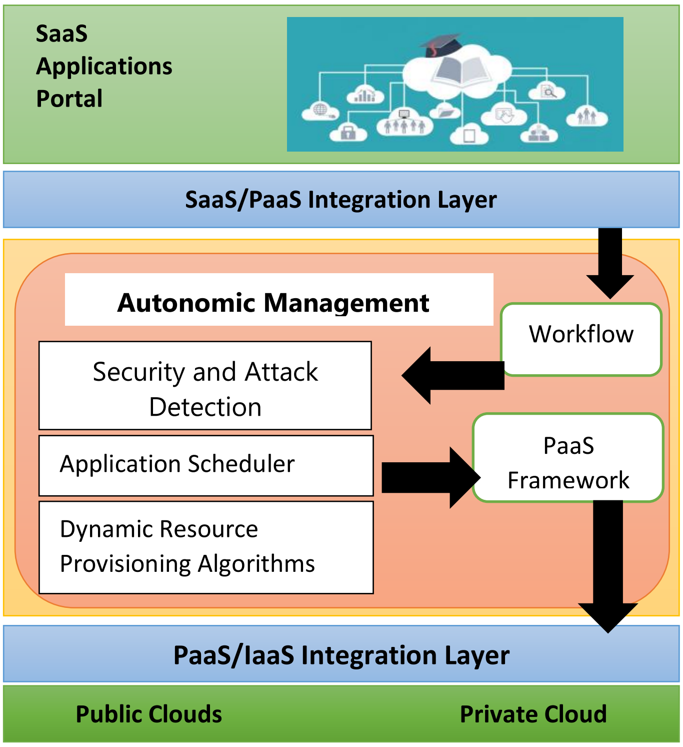
SaaS
Applications
Portal
SaaS/PaaS Integration Layer
Autonomic Management
Workflow
Security and Attack Detection
Application Scheduler
PaaS Framework
Dynamic Resource Provisioning Algorithms
PaaS/IaaS Integration Layer
Public Clouds
Private Cloud

Fig.4 Autonomic Cloud Management Architecture

C. Autonomic Cloud Computing and Match making Technique

The goal of Match Making technique is effective strategies for achieving the highest Quality of Service (QoS) in particular the Service Level Agreements, i.e (SLA). A typical SLA life cycle involves three main phases, Negotiation, Monitoring and Enforcement. In the *Negotiation* phase, customer(s) and provider(s) negotiate on requirements/services to agree on what the SLA should offer effectively. In the *Monitoring* phase, an agreed upon SLA is checked for its actual degree of conformance or for penalties if violated. Monitoring includes (a) verifying that the SLAs are respected, and (b) generating alerts whenever SLAs are not fulfilled. In the *Enforcement phase*, the actions needed to comply with the SLA (i.e., to keep a sustained level of security) are taken. The three phases are interlinked: the Negotiation cannot be performed without considering how SLA can be granted, Enforcement needs Monitoring in order to evaluate the real state of the solution before applying the predefined policies and procedures, while Monitoring needs the results of negotiation to know what to monitor and which alerts should be generated.

The Match Making Technique takes into consideration the heterogeneity of hardware resources, the uncertainties regarding the scheduling policies at the resources, and the dynamic nature of the resources. Fig 5. Shows the Resource Management Framework for Match Making technique.

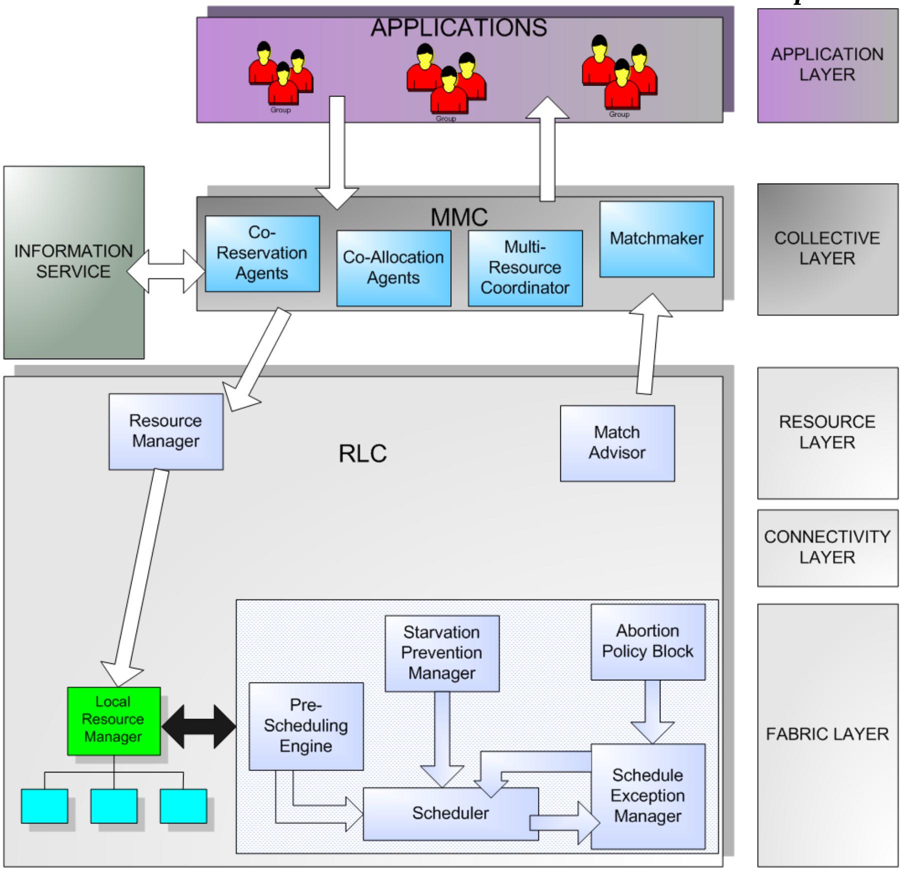

Fig. 5 Resource Management Framework for Match Making Technique

D. Resource Provisioning using Match making Technique

The Resource provisioning using Match Making technique involves two steps. 1) Matching 2) Scheduling. Matching is the process of procuring resources and assigning a job to that particular resource. Scheduling determines the order in which the assigned jobs are executed in a resource [10].

The Algorithm employs scaling using subset scheduling. This method allows the algorithm to screen the incoming requests to ascertain whether the new request will anyway hinder the execution of other tasks already in the resource schedule and tries to work out a feasible schedule involving only the subset of tasks.

E. Match making Algorithm

The cost of a resource provisioned depends on the time duration for which it is used and the type of the resource. In a higher educational institution's cloud, students who request for services can be prioritized into two levels. i.e Basic and Advanced. Basic users are those who avail services for a shorter duration of time and Advanced users are those who avail services for a longer duration of time. The price of basic services is planned to be low compared to the advanced. Higher priority can be accorded to the advanced users based on supply and demand. During peak times, users can be priced slightly higher irrespective of the priority. The algorithm uses match making to manage the supply and demand.

Step 1: Initiate the cloud resources

Step 2:Admit user requests and check user credentials. If valid user, then process the request, else reject the request.

Step 3: Obtain the resource schedule set and check, if the new request will anyway disturb the execution of already admitted tasks and vice-versa

Step 4: Analyze whether downgrading the public Cloud instance type still enable completion of the workflow within the same time slot. If so, utilize the simpler and cheaper instance type.

Step 5: Obtain an initial solution for tasks in the Resource Schedule set using the modified Earliest-Deadline-First strategy that accounts for both preemptable and non-preemptable tasks. If not feasible, reject the new task.

Step 6: Run the accepted tasks on the schedule nodes.

Step 7: Check the SLA parameter and monitor the allocated resources and availability of resources. If supply > demand, maintain status quo, else make changes in resource allotment by provisioning more virtual machines.

Step 8: Check the status of allocated resources. If usage time > threshold value & demand > threshold value, terminate the service. Calculate the price, by subtracting the penalty charge for abrupt termination. Update the resource schedule set.

Step 9. Release resources which have completed the task and check whether SLA parameters are met. Calculate usage time and price. Update resource schedule set.

Step 10: Check demand for resource requests. If request pattern low and demand low, terminate virtual machines based on existing requests and patterns.

F. Advantages of the Match Making Algorithms

- Since the algorithm uses a two stage process, it is possible to optimize the resource provisioning and bring down the cost involved.
- Since the provisioning model is constantly updated based on the request patterns, the resources are scaled up or down according to the user requirements, thereby ensuring SLA.
- Truly a pay-per-use model, as pricing includes penalty for incomplete requests. Penalty is a SLA component.
- QoS parameters are ably met, as customer satisfaction is expected to be high. Customer Satisfaction is also a SLA component.
- Although provisioning of resources is a NP hard optimization problem, the proposed algorithm in the model provides an efficient solution to handle the resources.

G. Results

The evaluation of the proposed algorithm was carried out in CloudSim tool. Deadline defined by the user is set as the SLA parameter. The experimental setup consisted of 5 static virtual machines, with the following configuration. 32 bit platform, 16 GB memory, 160 GB of instance storage and Rs.50 per instance per hour. For the experiment a CPU intensive application is used. Each job consisted of 100 tasks and the execution time of each task was set as 3 minutes. Initially, the job was executed sans SLA parameter and then the SLA parameter was included. The deadline was changed for each iteration. The deadlines used in the experiment are 15 mts, 30 mts and 45 mts. Table 1 depicts the results of the experiment.

	Initial static machines	**Machines provisioned as per the needs**	**ARPCM**	
			Job execution time	Extra Price
Without QoS	5	0	1:53:41	Rs. 0
45 mts	5	2	1:08:53	Rs.18
30 mts	5	6	0:52:05	Rs. 56
15 mts	5	20	0:43:53	Rs.157

The results show that, the QoS requirements are efficiently met by the algorithm. To meet the user requirements, the resource management model was able to provision more number of virtual machines and finish the execution within the stipulated time. Also, as the user requirement goes up, so too the machines provisioned, thereby meeting the SLA.

IV. CONCLUSION AND FUTURE DIRECTIONS

The growing trend of cloud adoption in higher educational institutions warrants new management models for this highly challenging computing environment. As the need for cloud computing solutions continue to grow exponentially, so too the threats. The proposed autonomic computing model is a humble first step in addressing the above problems. The dynamic nature of autonomic provisioning is found to be able to satisfy the QoS (Quality of Service) requirements of the consumer, resulting in improved efficiency in managing resource provisioning and optimized consumption of energy. Also, the proposed algorithm employs a two stage Match Making process, employing scaling through subsets This helps the model to identify malicious requests and thereby prevents DDoS attacks. Also, it does away with wastage of budget and energy consumption due to these malicious requests. The SLA obligations are thoroughly met, thereby resulting in increased customer satisfaction. In future more QoS aware algorithms could be employed resulting in more energy consumption and security aware architecture could be realized.

REFERENCES

[1] Rajkumar Buyya, Rodrigo N. Calheiros, and Xiaorong Li, —Autonomic Cloud Computing: Open Challenges and Architectural Elements‖, *IEEE Digital Library*, EAIT, 2012.

[2] J.O. Kephart and D. M. Chess, The Vision of Autonomic Computing, *Computer*, 36(1):41-50, IEEE, Jan. 2003.

[3] K. Appleby, S. Fakhouri, L. Fong, G. Goldszmidt, Kalantar, S. Krishnakumar, D. P. Pazel, J. Pershing, and Rochwerger. —Oceano-SLA based management of a computing utility‖. *Proceedings of the 2001 IEEE/IFIP International Symposium on Integrated Network Management (INM 2001), Seattle, WA, USA*, 2001.

[4] R. Buyya, C. S. Yeo, and S. Venugopal,‖.Market oriented Cloud Computing: Vision, Hype and Reality for delivering IT services as computing utilities‖, *Proceedings of 10th International Conference on High Performance Computing*, 2008.

[5] H. Liu, V. Bhat, M. Parashar and S. Klasky,‖ An Autonomic Service Architecture for Self-Managing Grid Applications‖, *Proceedings of the 6th IEEE/ACM International workshop on Grid Computing*,USA,2005

[6] R. Ranjan, R. Buyya, and M. Parashar, —Autonomic Cloud Computing: Technologies, Services, and Applications, Concurrency and Computation: Practice and Experience‖, 24(9):935-937, Wiley Press, New York, USA, June 2012.

[7] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P.Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. OceanStore: an architecture for global-scale persistent storage, *ACM SIGPLAN Notices*, 35(11):190-201, ACM, Nov. 2000.

[8] J. Menon, D. A. Pease, R. Rees, L. Duyanovich, and B. Hillsberg. IBM Storage Tank—A heterogeneous scalable SAN file system, *IBM Systems Journal*, 42(2):250-267, IBM, Apr. 2003

[9] Mayanka Katyal and Atul Mishra, Application of Selective Algorithm for Effective Resource Provisioning In Cloud Computing Environment, *International Journal on Cloud Computing: Services and Architecture (IJCCSA) ,Vol. 4, No. 1,* February 2014

[10] G. M. Lohman and S. S. Lightstone, SMART: making DB2(more) autonomic. *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB 2002),* Hong Kong, China, 2002

[11] H. Kim, Y. el-Khamra, I. Rodero, S. Jha, and M. Parashar, Autonomic Management of Application Workflows on Hybrid Computing Infrastructure, *Scientific Programming* 19(2):75-89, IOS, June 2011.

[12] J. Kaufman, T. Lehman, G. Deen, J. Thomas, OptimalGrid -autonomic computing on the Grid, IBM, 2003.

[13] J. Chen, C. Wang, B. Zhou, L. Sun, Y. Lee, A. Zomaya,Tradeoffs Between Profit and Customer Satisfaction for Service Provisioning in the Cloud, *Proceedings of the 20th* ACM International Symposium on High Performance Distributed Computing (HPDC 2011), San Jose, USA, 2011

[14] A. Goscinski and M. Brock, Toward dynamic and attributebased publication, discovery and selection for Cloud computing, *Future Generation Computer Systems*, 26(7): 947-970, Elsevier, Jul. 2010.

[15] B. Bethwaite, D. Abramson, F. Bohnert, S. Garic, C.Enticott, T. Peachey, Mixing the Grid and Clouds: Highthroughput Science using the Nimrod Tool Family, In: *Cloud Computing: Principles, Systems and Applications,*Antonopoulos and Gillam (eds), Springer, 2010..

[16] Sleiman Rabah, May El Barachi, Nadjia Kara, Rachida Dssouli and Joey Paquet A service oriented broker-based approach for dynamic resource discovery in virtual networks, *Rabah et al. Journal of Cloud Computing: Advances, Systems and Applications,* (2015) 4:3

[17] Lei Ying,Siva Theja Maguluri and R. Srikant,Heavy Traffic Optimal Resource Allocation Algorithms for Cloud Computing Clusters, *dl.acm.org,* 2012

[18] K. Appleby, S. Fakhouri, L. Fong, G. Goldszmidt, M. Kalantar, S. Krishnakumar, D. P. Pazel, J. Pershing, and B.Rochwerger. Oceano-SLA based management of a computing utility. Proceedings of the 2001 IEEE/IFIP International Symposium on Integrated Network *Management (INM 2001),* Seattle, WA, USA, 2001.

[19] Swarupa Irugurala, Dr.K.Shahu Chatrapati, Various Scheduling Algorithms for Resource Allocation In Cloud Computing, *The International Journal Of Engineering And Science (IJES), Volume 2, Issue 5,Pages 16-24,2013*

[20] Zhen Xiao, Weijia Song, and Qi Chen, Dynamic Resource Allocation using Virtual Machines for Cloud Computing Environment, *IEEE TRANSACTION ON PARALLEL AND DISTRIBUTED SYSTEMS. VOL. 24 , NO. 6* YEAR 2013,

[21] Hadi Goudarzi and Massoud Pedram, Multi-dimensional SLA- based Resource Allocation for Multi-tier Cloud\Computing Systems, *ACM digital Library,* 2012

[22] Mansoor Alicherry, Network Aware Resource Allocation in Distributed Clouds, *Proceedings IEEE INFOCOM,*2012.

[23] Brett Snyder1, Jordan Ringenberg, Robert Green, Vijay Devabhaktuni and Mansoor Alam1Evaluation and design of highly reliable and highly utilized cloud computing systems, Snyder *et al. Journal of Cloud Computing Advances, Systems and Applications* (2015) 4:11

AUTHOR PROFILE

Prof. G. Sobers Smiles David, is working as Asst., Professor and Head, PG Department of C.S, Bishop Heber College (Autonomous), Trichirappalli. He has got 15 years of U.G and P.G teaching experience. He has got his M.C.A from Bishop Heber College in 2001. He has got his M.Phil in C.S from Bharathidasan University in 2004. Currently he is pursuing Ph.D in Bharathidasan University. His areas of interest include Cloud Computing, Computer Networks, Cryptography and Network Security, Pervasive Computing, and distributed computing.

Dr. R. Anbuselvi is working as Asst., Professor, PG Department of C.S, Bishop Heber College (Autonomous), Trichirappalli. She has got 20 years of UG and PG teaching experience. She has been awarded Ph.D in Computer Science through Mother Teresa Women’s University, Kodaikkanal. Her research interest includes Artificial Intelligence, Data mining and Warehousing, Distributed Systems and Cloud Computing.