An Efficient Framework for Resource Allocation in Cloud Computing

Aman Kumar, Emmanuel S. Pilli and R. C. Joshi
Department of Computer Science & Engineering
Graphic Era University, Dehradun, India
{aman.kec, emmshub, chancellor.geu}@gmail.com

***Abstract*—Presently Cloud Computing is on high demand as it provides a way to reduce the cost of building infrastructure through virtualization of resources. Virtualization of resources requires a highly dynamic resource management mechanism. As cloud computing provides the facility to the cloud users to send multiple request simultaneously, there must be a self managing/ provisioning scheme that all resources are made available to the requesting users in the efficient manner to satisfy their requirement and for improvement of resource utilization. In this paper we proposed an efficient framework named called EARA (Efficient Agent based Resource Allocation) for resource allocation based on agent computing on SaaS level in Cloud Computing. EARA Contain five different agents, each agent equipped with functionality to collect information regarding all resources available in actual cloud deployment based on signed SLA agreement, and then replies to the user with appropriate allocation or response code.**

Keywords: Resource Allocation, SLA, Cloud Computig, EARA.

I. INTRODUCTION

National Institute of Standards and Technology (NIST) defines Cloud computing as follows: *"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"* [1]. This Cloud model promotes availability and has five essential characteristics including On-demand self-service, Broad network access, Resource pooling, Rapid elasticity and Measured Service, three service models including IaaS, PaaS and SaaS [2] where SaaS is a service provided to client in terms of applications running on the cloud computing infrastructure hosted by the service providers. PaaS refers to services which provide high-level integrated environment to design, build, run, test, deploy and update the applications created by client using development language and tool say Java, Python, .Net etc. provided by the service providers to the cloud infrastructure. IaaS refers to the services provided to the users is to lease the processing power, storage, network and other basic computing resources, with which users can deploy and run any software including operating systems and applications, Four deployment models including the private cloud, the public cloud, the hybrid cloud and the community cloud.

Cloud computing is basically a resource that you can utilize online to maintain your business's platform while you worry about other basic assets about your business. It basically changes how online developers meet their criteria for their IT software. Virtualization is an important feature in cloud technologies, which makes separation from software in hardware. Users can run their own applications in the' Virtual Machines', without caring the background jobs. Thus cloud computing, will help organizations be released from the pressure of upgrading the hardware and software [3].

Cloud Computing is coupled with a new paradigm for the provision of computing infrastructure. This paradigm shifts the location of this infrastructure to the network to reduce the costs associated with the management of hardware and software resources. The Cloud is drawing the attention from the Information and Communication Technology (ICT) community, thanks to the appearance of a set of services with common characteristics, provided by important industry players. At any rate, some of the existing technologies the Cloud concept draws on (such as virtualization, utility computing or distributed computing) are not new [4].

Cloud Computing is a model of service delivery and access where dynamically scalable and virtualized resources are provided as a service over the Internet. The overarching goal of Cloud Computing is to provide on-demand computing services with high reliability, scalability, and availability in distributed environments. One of the important requirements for a Cloud computing environment is providing reliable QoS. QoS can be defined in terms of Service Level Agreements (SLA) that describes such characteristics as minimal throughput, maximal response time or latency delivered by the deployed system [5].

All the QoS constraints, like throughput, response time and power consumption, are mainly dependent on the mechanism of efficient and effective resource utilization. The effective resource utilization is directly derived from fine grained efficient resource allocation mechanism. The cloud computing is the concept of provide the virtualized resources to the multiple user's at a time. Hence the cloud computing environment is also required a powerful self adaptive resource mechanism to make it reliable.

In this paper we proposed an efficient framework for the Resource Allocation in Cloud Computing named called EARA (Efficient Agent based Resource Allocation). EARA Contains five different agents, each equipped with functionality to collect information regarding all resources available in actual cloud deployment, based on signed SLA agreement, and then replies to the user with appropriate allocation or response code. We discussed essentials of Cloud Computing in section II as Related Work, in Section III we depicts our efficient model for resource allocation, in section IV we described Vector Space model with respect to resource allocation section V concluded with future directions.

II. Related Work

Resource allocation is one of the most challenging problems in high performance computing field for managing and provides effective utilization of resources. This problem has concerned a lot of concentration from the research groups in the last few years. We give a review of most important earlier effort.

Emeakaroha et al. in [6] presented an efficient resource manager for dynamic virtualized resources allocation for cloud computing environment. This anticipated mechanism consists of three components: User Interface (UI), Subscriber Server (SS) and Resource Manager (RM). User can access cloud in authorized manner with corresponding user name and password, through user interface. Resource manager accepts VMs request from user interface and subscription message to Subscriber server. The responsibility of Subscriber server is to maintain and manage the user profiles and then resource manager manages the resource distribution.

Chang et al. in [7] proposed an approximation algorithm based on their problem findings that is how many and what kinds of resources require to be reserved from clouds by formulating demand for computing power and other resources. Author treats it as a resource allocation problem with multiplicity. In this concurrent computation are represented as tasks and a later task can reuse resources released by an earlier task. The author also depicts it as NP-Complete problem to minimize the cost/size of the reservation of resources. The algorithm can also be used by both enterprises cloud providers to reduce cost or to allow a higher degree of resource sharing.

Van et al. [8] purposed management system relies on two-level architecture with a clear partition between application-specific functions and generic global decisions. They used utility functions to map the current state of each application like workload, resource capacity and SLA to a scalar value that quantifies the ”satisfaction” of each application with regard to its performance goals. These utility functions are also communicate with the global decision layer which constructs a global utility function that includes resource management costs and divide the VM provisioning stage from the VM placement stage. Both problems are instances of an NP-hard knapsack problem for which author suggest Constraint Programming approach to use i.e. to solve a problem by stating relations between variables in the form of constraints which must be satisfied by the solution.

Urgaonkar et al. in [9], investigate optimal resource allocation and power management in virtualized data centers with time-varying workloads and heterogeneous applications. The author makes use of Lyapunov Optimization [10] to design an online admission control, routing, and resource allocation algorithm for a virtualized data center. This algorithm makes use of the queuing information available in the system to implicitly learn and adapt to unpredictable changes in the workload and does not require estimation and prediction of its statistics. The technique of Lyapunov Optimization has been used to develop throughput and energy optimal cross-layer control algorithms in time varying wireless networks.

Zhong et al. in [11], a scheduling algorithm based on Improved Genetic Algorithm (IGA) is proposed for the automated scheduling strategy. The automated scheduling model is being divided into three steps. First, the scheduler updates the available resource list when allocation or de-allocation happens and update the VM request list when each time new VM requests come. Second, the scheduler uses an IGA algorithm to get out a fitness and economical allocation and final step, the cloud launches the corresponding VMs at the physical resource and suspends the VMs when the leasing time’s over.

Yazir et al. [12], Proposed a new approach to the dynamic resource based on Multiple Criteria Decision Analysis. The system is designed as a distributed network of NAs, each capable of accommodating VMs. Delegating VMs to other PMs and handing the management over to the corresponding NAs is also considered, when required. NAs carry out configurations through Multiple Criteria Decision Analysis (MCDA). The network maintains global awareness of the resource availability of PMs is the pre-assumption here, and their tasks are assigned through an oracle. The CoreASM is an open source tool environment for modeling dynamic properties of distributed systems. Hence, the system model is described in abstract functional and operational terms based on the Abstract State Machine (ASM) structure.

Wei et al. in [13] presents a game theoretic method to plan reliant computational cloud computing services with time and cost constrained. An evolutionary design is made to reasonably and approximately solve the NP-hard scheduling problem by introducing the nash equilibrium game-theory. Author considered resource allocation problem as QoS-constrained where service demanders intend to solve sophisticated computing problem by requesting the usage of resources across a cloud based network, and a cost of each network node depends on the amount of computation and also focused on the parallel tasks allocation problem on unrelated machines connected across internet. The concept of Nash Equilibrium is in some sense the analog of centralized optimal design in the context of multiple distributed selfish participants.

Head et al. in [14], the virtual Hypervisor abstraction cloud model is proposed, to make powerful solution managers to have much better control over the resource allocation for its specific VMs. By using the Virtual Hypervisor model the solution manager cause about locality, resource allocation, and workload allocation within its environment while at the same time the cloud manager can reason about the physical resource utilization and allocation and make decisions which optimize resource usage while not compromise the hosted solution. It’s provides a fine grained separation between both hosted as well as cloud base infrastructure management.

Wang et al. in [15], proposed an adaptive resource management approach. They consider two things multi-resource transformation to fully utilize extra resource capacity, also resource co-allocation issues of virtualized data centers in the cloud environments. An application environment (AE) consists of several virtual machines (VMs) distributed upon separate underlying PMs. The VMs cannot be arbitrarily deployed onto PMs in the considered data center architecture.

The main considerations includes three things, co-allocation of multiple kinds of resources, dynamic and adaptive resource co-allocation strategies and step-wise optimization for multiple round in each interval, means the algorithm repeatedly optimizes the VM placement and the capacity allocation, which ensures that the load variation could be captured periodically.

Goudarzi et al. in [16] proposed a force-directed search algorithm to solve SLA-based resource allocation problem for multi-tier applications in the cloud computing environment. Author also provides an upper bound to the algorithm on the total profit. The optimization is done over the three dimension processing, memory requirement, and communication resources. Simulation results demonstrate the effectiveness of the proposed heuristic algorithm.

Jung et al. [17] proposed an adaptive resource allocation model that allocates the consumer's job to a suitable data center. The model is implemented in an agent based test bed. The algorithm is provided to adaptively find a proper data center is mainly based on two measurements, 1) the geographical distance, derived from network delay between a user and data centers, and 2) the workload of each data center. An agent based which contains User, Coordinator and Monitoring Agents; test bed was designed and implemented to demonstrate the proposed adaptive resource allocation model. The test bed was implemented using Java Agent Development framework (JADE).

Haresh et al. in [18] proposed an agent based model to improve the resource allocation in federated cloud. Resources will have to be collected from different providers, which is a complex procedure. The proposed model has three types of agents namely Consumer agent, Resource Brokering agent and Resource Provider agent. According to author in this method, the user needs not know who the cloud service provider is and where the resources reside. The consumer gets the resources with the minimum price and administrative overhead.

III. Vector Space Computation

The Vector Space Model is an information retrieval strategy that computes a measure of similarity by defining a vector represents each documents D1, D2,…., Dn and another vector for representing the query(Q). Vector Space model based on identifies the Similarity Coefficient SC (Q, Di) of each document $1 \le i \le n$ [19].

This model involves constructing of vectors that represents the terms. Hence, in our framework we can also represent customer query (provided by SLAA-Service Level Agreement Agent, after checking agreement database with cloud provider) regarding demand of resources as one vector CRQ (Customer Resource Query). The vector PAR (Provider's all Resources) can be taken for the provider's declaration of total resources over the respective portal.

Similarity Coefficient (SC) can be calculated similarly between both vectors, CRQ and PAR, time-to-time as new customer demands for resources according to their SLA agreement. The vector formation scenario can be shown as given in figure 1.

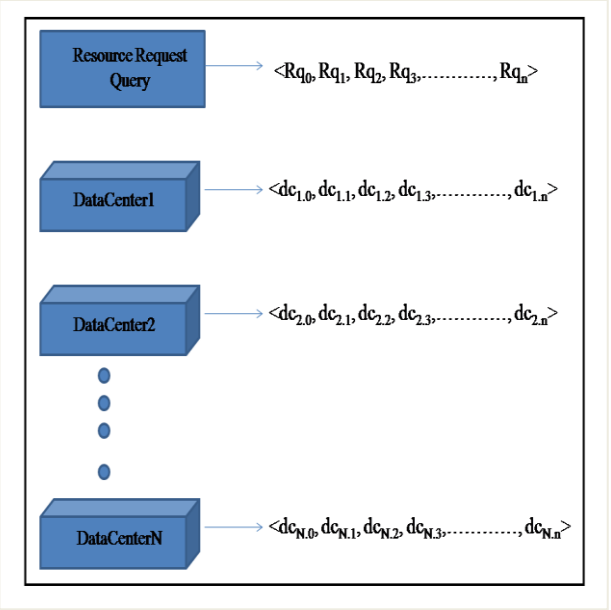

Fig. 1. CRQ and PAR Vector Creation

According to the Vector Space Model, we have to set the following parameters to compute the Similarity Coefficient (closeness of particular resource type and DataCenter).

Rt= Number of distinct resources in the DataCenters.

RtO_{ij} =Number of available resource of a particular type in DataCenter dc_i. (This information provided by CSCA Agent after checking the current status log of each resource type)

dcO_j =Numbers of DataCenters which contain Rt_j (DataCenter Frequency).

$idcO_j = \log(dc_N/dcOj)$, Where dc_N is the total number of DataCenters and $idcO_j$ is Datacenter's inverse frequency.

The vector for each DataCenter has n components including an entry for each distinct resource type. The components in the vector are filled with weights computed for each resource type in the DataCenters. The resources in each DataCenters are automatically assigned weights based on how much (in numbers) they available in the entire DataCenters Collection and how often a particular resource type available in a particular DataCenter. The weight of a particular resource type in DataCenter increases the more often the resource available in one DataCenter and decreases the more often it available in all others DataCenters. A weight computed for a resource in a DataCenter vector is non-zero only if the resource available in the Datacenter.

The weighting factor for a resource in a DataCenter is defined as a combination of Resource Availability Frequency and Inverse Datacenter Frequency, means if we want to compute the value of the jth entry in vector corresponding to $DataCenter_i$,

$$dc_{ij} = RtO_{ij} * idcO_j$$

A simple similarity coefficient (SC) between customer query (put variable for query here) and a DataCenter dc_i can be defined by the dot product of two vectors.

$$SC(CRQ, dc_i) = \sum_{j=1}^{Rt} w_{qj} * dc_{ij}$$

IV. The EARA Framework

The powerful self adaptive resource allocation mechanism is directly implies the capability of the system to serve a large number of users simultaneously. In cloud computing environment, the pools of virtualized resources are provided to serve, but, the final decision to allocate a particular resource is taken by the resource allocation mechanism to make the full utilization of resource. Here we gives, Efficient Agent based Resource Allocation, framework named called EARA.

EARA Contains five different agents, each equipped with functionality to collect information regarding all resources available in actual cloud deployment, based on signed SLA agreement, and then replies to the user with appropriate allocation or response code. The layered architecture for EARA given as follows.

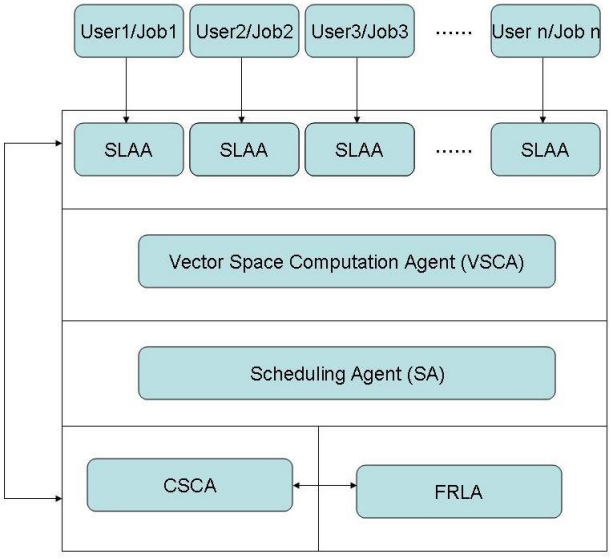

Fig. 2. EARA Layered Architecture of Cloud Resource Allocation Model

A. Cloud Interaction Layer

CSCA (Current Status Computation Agent): The current status computation agent contains the code to maintain information of current status of the particular/all resources, the out information then further used in computation of the appropriate agent.

FRLA (Fast Resource Locator Agent): This agent equipped with the technique which provides the set of resources which is appropriate with the customer agreement and used as one vector of input for the VSCA for calculating the best one with optimized response time by following the all performance metrics as discussed above.

B. Scheduling Layer

SA (Scheduling Agent): The development and operation performed by this agent is the main work of the proposed model. The Scheduling Agent devised with optimized priority based scheduling algorithm, the major work is the allocation of the resources based on the information provided by the several agents and computation result provided by VSCA (Vector Space Computation Agent).

C. Resource Computation Layer

VSCA (Vector Space Computation Agent): This agent equipped with the algorithm of Vector Space Information Retrieval technique, its taking to set (vector) as input i) Vector of Resources: collected from FRLA (Fast Resource Locator Agent), which is working compositely with CSCA (Current Status Computation Agent), to update the information of available resources periodically ii) Vector of Request: collected from SLAA (Service Level Agreement Agent), after the complete confirmation of the agreement with the customer to the service provider.

D. Customer Interaction Layer

SLAA (Service Level Agreement Agent): On the basis of the customer agreements with the service provider and also the information provided by CSCA (Current Status Computation Agent), this agent have accountability of the service resources, service, service time and cost of particular usage.

V. Implementation

To implement and validate the proposed framework functionality initially, we code this framework in JAVA on JADE (Java Agent Development Environment), and we have also created a small Eucalyptus based private cloud-in-box setup of single system. The machine which hosts the private in-box cloud is heaving configuration, a minimum of 200GB disk space and 4GB RAM to deploy the Eucalyptus FastStart private cloud package with CentOS 5 and KVM by using release Eucalyptus 3.2.1 FastStart[20]. The current implementation block view is shown in figure 3.

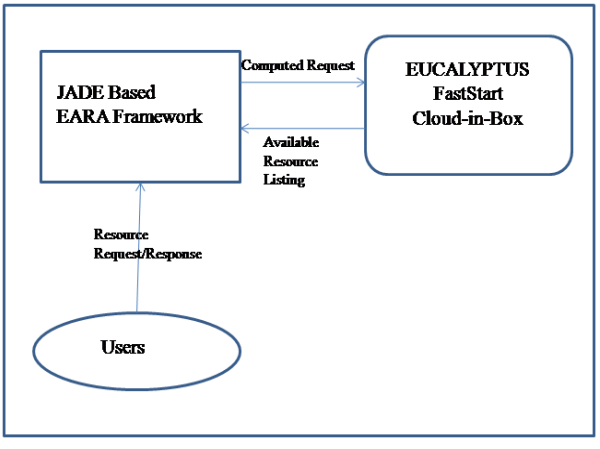

Fig. 3. A general block diagram of current implementation

The activity block diagram based on different agents communication, shown in figure 4.

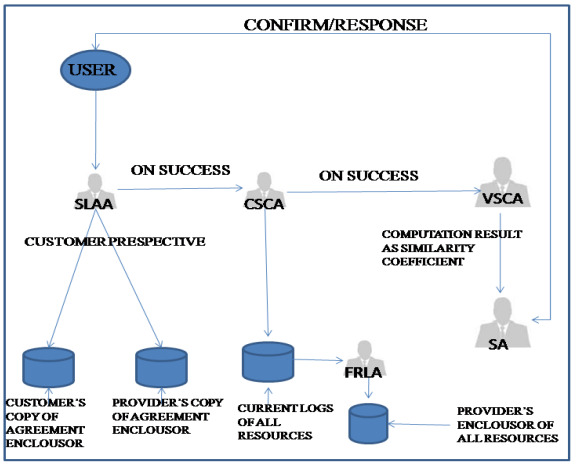

Fig. 4. Agent Communication Block Diagram

All agents used ACLmessages to communicate each other for receiving and sending the information from database or from other agent. The communication flow, as shown in above figure 4, start from customer's login with resource request. The request query is formatted into an input vector by extracting the names of actual demanded resources.

The SLAA (Service Level Agreement Agent) got initial query string from user interface of provider's portal. Then through ACLmessages(), extracted query of resources provided to CSCA (Current Status Computation Agent) to check the availability and the current status of the demanded resources. The final result from CSCA and extracted query are provided to VSCA (Vector Space Computation Agent) as input to calculate the similarity coefficient (SC) between demanded resources and the availability of resources in datacenters, as the periodic result provided by the CSCA agent whenever a resources is demanded by user.

If the customer request CRQ= "R_1 R_2 R_3" and if the scenario of total available resources (PAR) at that time is

$DataCenter_1\,(dc_1) = "R_2\,R_3\,R_4\,R_5"$

$DataCenter_2\,(dc_2) = "R_2\,R_2\,R_3\,R_4\,R_5"$

$DataCenter_3\,(dc_3) = "R_1\,R_2\,R_3\,R_3\,R_3\,R_4"$

Hence the total number of DataCenter $dc_N = 3$, and we can calculate $idcO_j = log\ (dc_N/dcOj)$, for all resources as given below

$idcO_{R1}=\log(3/1) = 0.477,$

$idcO_{R2}=\log(3/4) = -0.126,$

$idcO_{R3}=\log(3/4) = -0.126,$

$idcO_{R4}=\log(3/5) = -0.221,$

$idcO_{R5}=\log(3/2) = 0.176,$

The Datacenter Vectors can be shown as in Table I.

TABLE I. DATACENTER VECTORS

$DataCenter_{id}$	R_1	R_2	R_3	R_4	R_5
dc_1	0	-0.126	-0.126	-0.221	0.176
dc_2	0	-0.252	-0.126	-0.221	0.176
dc_3	0.477	-0.126	-0.378	-0.221	0
CRQ	0.477	-0.126	0	0	0.176

The similarity coefficient can be calculated according to the VSM model, for the above scenario the calculated SC (Similarity Coefficient) shown in Table II.

TABLE II. SIMILARITY COEFFICIENT

$DataCenter_{id}$	$SC(CRQ, dc_i)$
dc_1	0.0151
dc_2	-0.0077
dc_3	0.2116

Hence, the DataCenter's allocation ranking can be defined as dc_2, dc_1, dc_3, means in this order a dynamic request can be served by the datacenters while it was the first request, to handle the more than one request, all agents required to be work simultaneously to update the status of particular request and also to update the total availability of resources at that moment.

VI. CONCLUSION AND FUTURE SCOPE

This paper gives the SLA aware Efficient Agent based Framework for Resource Allocation on SaaS level for several Cloud Computing Providers. Resource Computation and Scheduling is done based on Vector Space Model of information retrieval. The Vector Space model computes the similarity coefficient based on two dimensional input set, one is the set of constraints set by the cloud provider and second the feature set subscribed by the customer at the time of service leasing and agreement with the service provider itself. A priority based (of similarity coefficient match) scheduling algorithm is then used by Scheduling Agent to allocate the resource to particular user. The framework can also be extended by having the capability of computing pricing policy and for energy aware resource management after enhanced optimization.

REFERENCE

[1] P. Mell and T. Grance, "The NIST definition of cloud computing (draft)," NIST special publication, vol. 800, pp. 145.

[2] N. Sadashiv, and S. M Dilip Kumar, "Cluster, Grid and Cloud Computing: A Detailed Comparison," in *6th International Conference on Computer*

Science & Education (ICCSE'11), Singapore, 2011, pp. 477- 482.

[3] Q. Shuai, "What will cloud computing provide for Chinese mlearning?," in *IEEE International Conference on e-Education, Entertainment and e-Management(ICEEE'11)*, Bali, 2011, pp. 171-174.

[4] L. M. Vaquero, L. R. Merino, J. Caceres, and M. Lindner, "A Break in the Clouds: Towards a Cloud Definition," in *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, 2009.

[5] B. Damien, M. Michael, D.-C. Georges, P. Jean-Marc, and B. Ivona, "Energy-efficient and SLA-aware management of IaaS clouds," in *3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet Madrid*, Spain: ACM, 2012.

[6] V. C. Emeakaroha, I. Brabdic, M. Maurer, and I. Breskivic, "SLA-Aware Application Deployment and Resource Allocation in Clouds", in *35th IEEE Annual Computer Software and Application Conference Workshops*, Munich, 2011, pp. 298-303.

[7] F. Chang, J. Ren, and R. Viswanathan, "Optimal Resource Allocation in Clouds," in *IEEE 3rd International Conference on Cloud Computing (CLOUD'10),* Miami, FL, 2010, pp. 418 - 425.

[8] H. N. Van, F.D. Tran, and J. M. Menaud, "SLA-aware Virtual Resource Management for Cloud Infrastructures," in *Ninth IEEE International Conference on Computer and Information Technology (CIT '09)*, Xiamen, 2009, pp. 357 – 362.

[9] R. Urgaonkar, U.C. Kozat, K. Igarashi, and M.J. Neely, "Dynamic Resource Allocation and Power Management in Virtualized Data Centers," in *IEEE International Conference on Network Operations and Management Symposium (NOMS'10)*, Osaka, 2010, pp. 479 – 486.

[10] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," in *Foundations and Trends in Networking*, vol. 1, no. 1, 2006, pp. 1-149.

[11] H. Zhong, K. Tao, and X. Zhang, "An Approach to Optimized Resource Scheduling Algorithm for Open-source Cloud Systems," in *Fifth Annual ChinaGrid Conference (ChinaGrid'10),* Guangzhou, 2010, pp. 124 - 129.

[12] Y.O. Yazir, C. Matthews, R. Farahbod, S. Neville, A. Guitouni, S. Ganti, and Y. Coady, "Dynamic Resource Allocation in Computing Clouds using Distributed Multiple Criteria Decision Analysis," in *IEEE 3rd International Conference on Cloud Computing (CLOUD'10),* Miami, FL, 2010, pp. 91 - 98.

[13] G. Wei, A.V. Vasilakos, and N. Xiong, "Scheduling Parallel Cloud Computing Services: An Evolutional Game," in *1st International Conference on Information Science and Engineering (ICISE'09),* Nanjing, 2009, pp. 376 – 379.

[14] M.R. Head, A. Kochut, C. Schulz, and H. Shaikh,"Virtual Hypervisor: Enabling Fair and Economical Resource Partitioning in Cloud Environments," in *IEEE Network Operations and Management Symposium (NOMS'10),* Osaka, 2010, pp. 104 - 111.

[15] X. Wang, H. Xie, R. Wang, Z. Du, and L. Jin "Design and Implementation of Adaptive Resource Co-allocation Approaches for Cloud Service Environments," in *3rd International Conference on Advanced Computer Theory and Engineering (ICACTE'10)*, Chengdu, 2010, pp. 484 -488.

[16] H. Goudarzi, and M. Pedram, "Multi-dimensional SLA-based Resource Allocation for Multi-tier Cloud Computing Systems," in *IEEE International Conference on Cloud Computing (CLOUD'11),* Washington, DC, 2011, pp. 324 - 331.

[17] G. Jung, and K. M. Sim, "Agent-based Adaptive Resource Allocation on the Cloud Computing Environment," in *40th International Conference on Parallel Processing Workshops (ICPPW'11),* Taipei City, 2011, pp. 345 – 351.

[18] M.V. Haresh, S. Kalady, and V.K. Govindan, "Agent Based Dynamic Resource Allocation on Federated Clouds," in *IEEE international Conference on Recent Advances in Intelligent Computational Systems (RAICS'10)*, Trivandrum, 2010, pp. 111 – 114.

[19] D.A. Grossman and O. Frieder, Information Retrieval: Algorithms and Heuristics, Springer, Dordrecht, Netherlands, 2004.

[20] "EUCALYPTUS", (2013), [Online]. Available: engage.eucalyptus.com/ [2013].