Enhanced and Optimal Resource Provisioning for Dynamic Service in Cloud Computing

[1]K. Phaneendra, [2]Dr. M. Babu Reddy
[1]*Research Scholar, Computer Science and Engineering, Rayalaseema University, Kurnool, Andhra Pradesh, India*
[2]*HOD, Department of Computer Science, Krishna University, Machilipatnam, Krishna, Andhra Pradesh India*

Abstract-Cloud computing, is an emerging concept to understand economics of distributed environment with critical and important for real time applications. To increase the utilization of cloud computing to outsource environment, cloud service provider should understand all services to determine different characteristics, or configurations. Optimal server configuration is required to increase maximization of different services in cloud computing. Traditionally use stochastic based resource provisioning prediction management (SRPPM) based on linear regression to satisfy resource provisioning and utilization in distributed computing. In this paper, we implement Optimal and Virtual Multiplexer Resource Provisioning (OVMRP) approach. Our approach reduces the resource provisioning cost by adjusting the on-demand and reservation instances to optimal. OVMRP treat a single server system for multiple users as an M/M/m queuing design, such that our optimization problem can be developed and fixed analytically. The probability density function of the waiting duration of a newly came support demand comes from. Performance evaluation of proposed approach can achieve high resource provisioning at less time and low usage of memory in cloud computing.

Keywords-*Cloud computing ,multiuser configuration, Cloud service provider, virtual machine and resource provisioning.*

I. INTRODUCTION

Cloud computing is a substantial scale disseminated figuring worldview in which a pool of registering resources are accessible to the clients by means of the Internet [2]. Processing resources, e.g., capacity, figuring force, stage, and programming, are spoken to clients as available administrations. Infrastructure as a-Service (IaaS) is a computational administration show connected in the distributed computing worldview [3]. Virtualization innovations can be utilized to help figuring resource access by the clients in this model. Clients can determine required programming stack, for example, working frameworks, programming libraries, and applications; at that point bundle them all together into virtual machines (VMs).

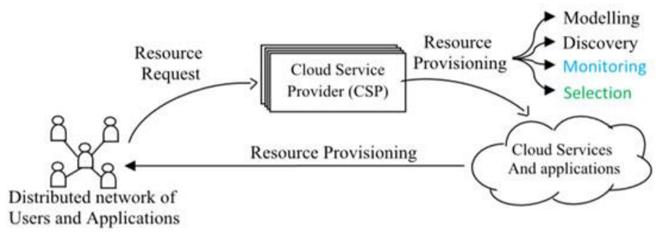

Fig 1: Resource Provisioning System in Cloud Computing (RPS)

At long last, VMs will be facilitated in a registering situation worked by outsider locales that we call cloud suppliers with efficient resource provisioning shown in figure 1. Cloud suppliers can offer clients two installment designs, i.e., reservation design (e.g., paid ahead of time) and on-request design (e.g., pay per utilize). Amazon EC2 [5] and GoGrid [6] are, for cases, the cloud suppliers which give IAAS administrations and offer reservation and on-request intends to the clients. By and large, cost of resources in reservation design is less expensive than that in on-request design. In any case, clients need to buy in a specific measure of resources in reservation design ahead of time for future use. Therefore, an under provisioning issue can happen when the measure of saved resources can't completely meet the requests. Luckily, this issue can be comprehended by buying in resources in on-request intend to fit the additional requests. Be that as it may, such on-request resources are all the more expensive, and the comparing cost is approached request cost. An over provisioning issue can't be neglected too since the measure of held resources will be underutilized. The cost of sit without moving held resource is by and large alluded to as finished provisioning or oversubscribed cost. Both on-request and oversubscribed costs should be limited. In this paper, limiting both under provisioning and over provisioning issues under the request and value vulnerability in distributed computing conditions is our inspiration to investigate a resource provisioning system for cloud customers. Specifically, Optimal and Virtual Multiplexer Resource Provisioning (OVMRP) is proposed to limit the aggregate cost for provisioning resources in a specific day and age. To settle on an ideal choice, the request vulnerability from cloud customer side and value vulnerability from cloud suppliers are considered to alter the tradeoff between on-request and oversubscribed expenses. Main contribution of our propose approach is as follows:

The ideal cloud resource provisioning calculation is proposed for the virtual machine administration.

A. The enhancement plan of stochastic number writing computer programs is proposed to acquire the choice of the Optimal and Virtual Multiplexer Resource Provisioning (OVMRP) all things considered the aggregate cost of resource provisioning in distributed computing conditions is limited. The detailing considers numerous provisioning stages with request and value vulnerabilities.

B. The arrangement techniques in view of Benders disintegration and test normal estimate calculations are utilized to explain the streamlining detailing in a proficient way.

C. The execution assessment is performed which can uncover the significance of ideal figuring resource provisioning. The execution examination among the Optimal and Virtual Multiplexer Resource Provisioning (OVMRP) and alternate methodologies is additionally displayed.

The proposed numerical investigation will be valuable to the cloud buyers (e.g., association and friends) for the administration of virtual machines in distributed computing condition. The proposed Optimal and Virtual Multiplexer Resource Provisioning (OVMRP) will encourage the reception of distributed computing of the clients as it can lessen the cost of utilizing registering resource fundamentally.

II. REVIEW OF RELATED WORK

In MUSE, every server has duplicates of all web programs working in the program. The conveyances deciding in a front side end L7-switch appraisals income are legitimately given while diminishing the measure of under-used web servers. Perform utilizes system stream figures to appropriate the weight of an application among its working occasions. For affiliation sorted out Internet organizations like Windows Stay Courier, work uncovers a synchronized methods for user resource provisioning and server provisioning. All works above don't utilize selective gadgets and require the projects comprising in a multi-level building plan with user resource shaping gave through a front-end dispatcher. Surprisingly, our work targets Amazon.com EC2-style climate where it puts no repression on what and how programs are created inside the VMs. A VM is taken care of like a black box. Source organization is done precisely at the granularity of entire VMs. Guide Decrease is another kind of predominant Reasoning organization where subtle elements zone is the heading to its execution. Quincy handles min-cost stream configuration in undertaking reservation to upgrade points of interest region while continuing inclination among various callings. The "Deferral masterminding counts bargains execution time for subtle elements zone. Perform dole out factor needs to occupations and clients to persuade resource assignment.

VM live development is an overall utilized system for factor resource part in a virtualized space. Our work similarly has a spot with these sorts. Sandpiper associates multi-dimensional bodyweight points of interest into a special Quantity estimation. It writes the quick overview of PMs in viewpoint amid their sums and the VMs in each PM in their volume-to-estimate level (VSR). This shockingly changed things way basic points of interest required when choosing the development choice. It by then perspectives the PMs and the VMs in the pre-arranged income request. We give a solid of the extra audit where their deciding decisions the wrong VM to move away amidst over-weight and can't moderate up the issue go. Different endeavors have been made to decrease vitality use in server plants. Segments based strategies be a piece of novel warm arrangement for bring down chilling vitality, or learning vitality relative and low-control equipment. Perform utilizes Powerful Volts and Regularity Climbing (DVFS) to give CPU vitality as affirmed by its fill. We don't utilize DVFS for green deciding the correlative record. Power Nap they depend on new equipment movements, for instance, Strong Condition Hard drive (SSD) and Self-Refresh DRAM to finish snappy transition(less than 1ms) between full capacity and low vitality state, with the reason that it can "rest" in short sit without moving spans. Resource provisioning alternatives were presented in [18].

Resource provisioning methodologies in circulated frameworks were tended. In, a probabilistic early booking was proposed. This reservation depends on existing best exertion bunch schedulers which again can't be ensured to be ideal. In [10], an idea of resource space was proposed. The goal is to handle vulnerability of resources accessibility. In [11], a double whole number program to amplify resource suppliers' incomes and use was figured. Heuristic strategies to fathom this twofold whole number program were proposed. Be that as it may, [11] – [15] did not consider vulnerability of future clients' requests. In [12], an enhancement structure for resource provisioning was produced. This structure considered numerous customer QoS classes under vulnerability of workloads (e.g., requests of processing resources). The landing example of workloads is evaluated by utilizing web based gaining strategies. Interestingly, our work indicates that requests given likelihood conveyances. At long last, these related works overlooked vulnerability of future requests and costs. What's more, the cost contrast amongst reservation and on-request designs was most certainly not considered. Stochastic programming [3] has been created to understand resource arranging under vulnerability. Stochastic programming was connected to take care of numerous issues in various fields, e.g., creation arranging, monetary administration, and scope organization. In any case, to the best of our insight, the application of stochastic programming to processing resource provisioning has never been considered.

III. BASIC PRELIMINARIES FOR OVMRP

OVMRP approach mainly defines resource provisioning based on virtual machines (VM) with respect to different services. Based on required virtual machines, define number of instances for resource provisioning. If service is required either it is on-demand or reservation instances then determine of stochastic integer programming can be formulated for virtual machine resource provisioning to minimize resources as follows

$$\min : \sum_{V_i \in V} \sum_{P_j \in p} c_{ij} X_{ij}^{r}$$

$$subjected : \sum_{P_j \in p} X_{ij}^{r} = v_i , V_i \in V$$

$$\sum_{V_i \in V} r_i^{(h)} X_{ij}^{(r)} \le t_j^{(h)} , P_j \in p$$

$$\sum_{V_i \in V} r_i^{(s)} X_{ij}^{(r)} \le t_j^{(s)} , P_j \in p$$

$$X_{ij}^{(r)} \in \{0,1,.....\}, V_i \in V, P_j \in p$$

The above equations are objective functions with decision variable $X_{ij}^{(r)}$ which defines number of virtual machines in particular class V with different variable and allocated provider P, c_{ij} denotes provisioning cost. Total provisioning cost may applicable with different services as follows:

$$c_{ij} = c_j^{(h)} r_i^{(h)} + c_j^{(s)} r_i^{(s)}$$

Virtual machine deployment for different virtual classes based on resource provider with decision variable, then resource provisioning for different users may changes from positive to negative at consistent situations.

Therefore stochastic integer programming for two stage resource provisioning is developed. In first phase, numbers of virtual machines are provisioned to use different services. Second stage defines actual VMs are require to provide services based on user consistent by providers. Stochastic programming can be represented as follows:

$$\sum_{V_i \in V} \sum_{P_j \in p} c_{ij} X_{ij}^{(r)} + \wp_{\Omega}[\wp(X_{ij}^{(r)}, w)]$$

$X_{ij}^{(r)}$ denotes number of VMs used for different users at first stage, $\wp(X_{ij}^{(r)}, w)$ defines number of VMs are required for resource provisioning. This procedure used to define stochastic integer programming for proposed approach i.e OVMRP at different user service utilization by service provider.

IV. SYSTEM MODEL AND IMPLEMENTATION

A. Design Procedure

Our proposed approach consist 4 stages in implementation i.e. Cloud service provider, users, virtual machine database and cloud brokers shown in figure 2.

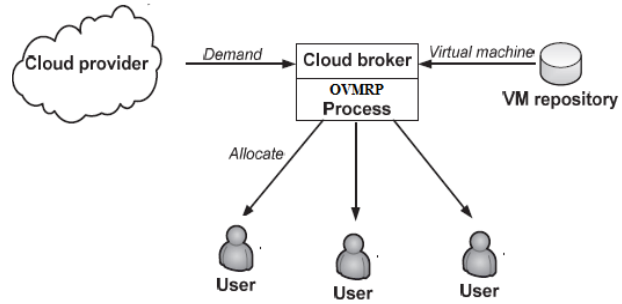

Fig.1 Proposed architecture for resource provisioning with different users.

As shown in figure 2, let us consider $v = \{V_1, V_2, ..., V_n\}$ be the set of instance virtual machines from first to last virtual data, in that single class defines different type of applications. For example, V1 be the mail server and V2 be the web server then combines these two servers using virtual machine. Same VM owner defines different classes for same web and mail servers. Let $p = \{P_1, P_2, ...P_n\}$, Pn be the last service provider of cloud, each cloud provider supplies sequential resources to each user. In this work, cloud provider computing energy consumption, storage and other specifications. The reasoning agent (Fig. 2) is a central enterprise (e.g, server) in the user’s website. The agent is responsible for providing VMs, saved in database, to reasoning suppliers. Furthermore, the OVMRP criteria (Fig. 2) are applied in the agent. This criterion is used to create maximum decision for the agent to a sources and variety VMs to any cloud suppliers.

We accept that each cloud supplier offers the client two installment designs, i.e., reservation and on-request designs. Cloud suppliers offer the cost of assets which will be charged to the client when the assets are held or used. Cost to arrangement assets in reservation design is thought to be less expensive than that in on-request design. There are three periods of provisioning assets: reservation, usage, and on-request. To begin with-in the reservation stage, without knowing client's request, the cloud agent arrangements assets in the reservation design. At that point, the use stage begins when the saved assets are utilized. Be that as it may, if the request surpasses the measure of held assets, the client can pay for extra assets in the on-request design, and at that point the on-request stage begins. In view of the three stages, there are three expenses related with provisioning assets: reservation, utilization, and on-request. Note that for a similar asset, a whole of reservation and usage costs is for the most part not exactly an on-request cost. The goal of the cloud representative is to limit every above cost while the request of clients is met. As previously mentioned, the cloud agent utilizes OVMRP to get an ideal arrangement. Truth be told, the ideal arrangement is to save the ideal number of assets in the reservation stage. An ideal arrangement is gotten by fathoming and detailing a stochastic number programming with two-organize response. There are two phases of basic leadership: first stage and second stage. The primary stage characterizes the quantity of VMs provisioned in reservation stage, while the second stage or plan of action characterizes the quantity of VMs assigned in both use and on-request stages. As such, the second stage speaks to the real number of VMs required by the client and real costs characterized by suppliers.

B. Algorithm Implementation

This section describes creation of resource provisioning for each user specification in cloud, and the process of resource provisioning shown algorithm 1.

```
1: procedure CREATERESOURCEPROVISIONINGPLAN(bot)
2:     if bot ∈ BoT_hom then
3:         solve MILP for homogeneous bot
4:         for each vmt that had at least one task assigned do
5:             numTasks = number of tasks assigned to a VM of type vmt
6:             numVMs = number of VMs of type vmt used
7:             RP_vmt = (numTasks, numVMs)
8:             RP_bot ∪ RP_vmt
9:         end for
10:    else if bot ∈ BoT_het then
11:        solve MILP for heterogeneous bot
12:        for each vm that had at least one task assigned do
13:            tasks = tasks assigned to vm
14:            RP_vm = (tasks, vm)
15:            RP_bot ∪ RP_vm
16:        end for
17:    else if bot ∈ BoT_sin then
18:        t = bot.task
19:        vmt_fast = find fastest VM that can finish the task within bot.budget
20:        if vmt_fast does not exist then
21:            vmt_fast = vmt_cheapest
22:        end if
23:        RP_bot = (vmt_fast)
24:    end if
25: return RP_bot
26: end procedure
```

Algorithm 1. OVMRP Step by step resource provisioning procedure for different services.

Algorithm 1 mainly consist 3 steps to schedule resource based on user request according to client's service request, first step is homogeneous based resource provisioning, second step is heterogeneous based resource provisioning and allocate service based on available resources. From line number 2-9, describes homogeneous service allocation if client service is present within same resource provisioning then update service server with number of tasks available at cloud service provider. From line number 10-16, if user selected service is not available at service provider then provisioning manager will get that service from equally matched service available in other resource provisioning server then allocate to user based on his budget. Remaining of algorithm steps describes allocate service based on service availability and start and ending of task within user budget. This is iterative process, whenever all the users complete their tasks within their budget and service availability.

C. Architecture

Figure 3 summarizes how the allowance and source arranging are mixed to obtain little energy intake in facilities nodes and hence data facilities. Both the actual Bin-Packing expansion and the Best-Fit heuristic are used to make sure maximum and sub-optimal positioning respectively.

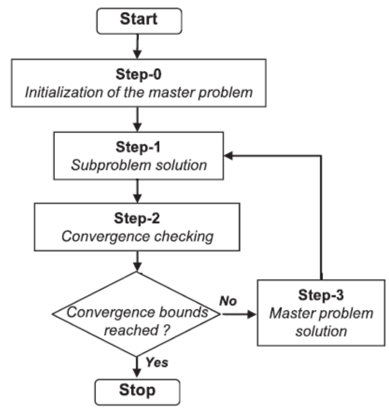

Fig.3 Design for resource allocation to multiple services.

The suggested methods are along with the migration with source arranging that is released if a variety of VM tasks cancel since their devoted sources become available for opportunistic recycling and for more efficient source allowance and submission. These departures are the chance of the merging criteria to change proportion by shifting VMs into the actual possible set of nodes. All purged or freed web servers (or nodes) are converted to reduce power consumption.

V. EXPERIMENTAL EVALUATION

In this section, we discuss proposed approach experimental set compare with traditional approach i.e stochastic based resource provisioning prediction management (SRPPM) in terms of memory, CPU and other resource parameters present in cloud computing. For that, we setup cloud environment using Net beans 8.0 and JAVA 8 using web designing and apache tomcat server to generate cloud deployment. Cloud parameters are giving as a data sets. Datasets with further dimensions were incapable to be examined on the test PC due to storage restrictions. Exams are conducted on an i5-3230M CPU with 2.60 GHz and 3 MB storage cache with 4 cores and 4 GB of RAM (3.86 GB is only usable). The PC operates MS Windows 7. The program was coded in Coffee. Datasets are produced by the program and stored to hard drive information.

We discuss classification accuracy with respect to different resources like CPU, Memory, Mean, Prediction Accuracy and Time analysis of proposed approach in terms of different services utilized by clients. As specified in above sections, we pre-process the data related cloud oriented resources to process different consistency client's data.
Results are appeared as follows:

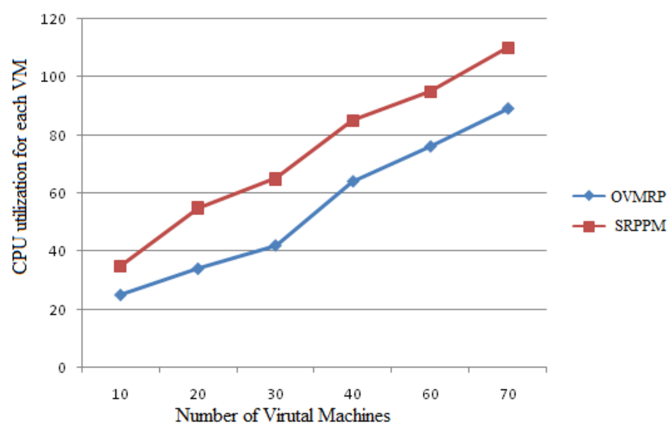

Fig.4 CPU utilization values for different virtual machine deployments.

This figure shows about utilization of different resources running on local host cloud with different services deployment.

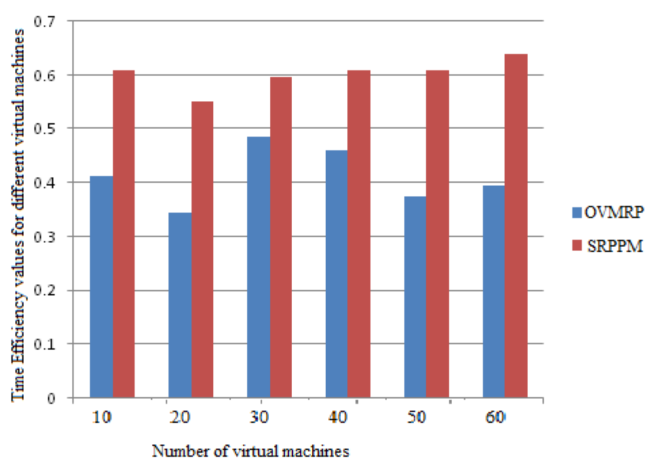

Fig. 5 Time efficiency values for different services.

Based on execution of different resource utilization with respect to different services shown in figure 5 with consistent resource utilization. Optimized cost results appeared in figure 6 with different virtual machines deployment with different services.

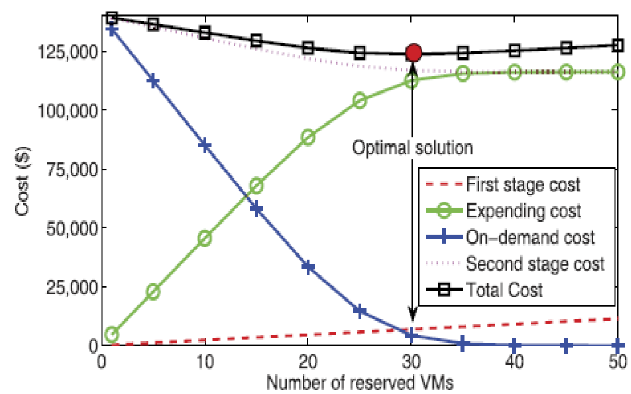

Fig.6 Optimal cost solution for different virtual machines.

In Fig. 6, given diverse number of saved VMs, cost in the principal arrange called first stage cost (which is really reservation cost), cost in the second stage called second stage cost including using and on-request expenses, and aggregate cost, are displayed. Obviously, the primary stage cost increments, as the quantity of saved VMs increments. Notwithstanding, the second stage cost diminishes after the request is acknowledged, since the cloud buyer needs more modest number of VMs provisioned by on-request design. For this situation, the ideal number of saved VMs can be resolved to be 30 saved VMs as appeared in Fig. 6, which is the point that the aggregate cost is least. Plainly, even in this little setting (one VM class and one supplier), the ideal arrangement isn't paltry to acquire because of the request vulnerability. In this way, the OVMRP calculation would be required to ensure the least cost to the cloud consumer. So that, the efficiency assessment of the OVMRP criteria has been done by mathematical studies and models. From the results, the criteria can optimally modify the compromise between booking of sources and allowance of on-demand sources.

VI. CONCLUSION

In this paper, we implement Optimal and Virtual Multiplexer Resource Provisioning (OVMRP) approach. Our approach is to treat a single server system for multiple users as an M/M/m queuing design, such that our optimization problem can be developed and fixed analytically. Our approach is mainly worked based on Infrastructure as a Service (IAAS) which define average visualization of resources for different services. OVMRP approach minimizes total resource provisioning cost in cloud computing environment, in this; we process different trade-off services like on-demand, reservation instances are adjusted to optimal. Our approach also do optimal to solve stochastic integer programming with two stage resource provisioning in cloud computing. Performance evaluation of proposed approach can achieve high resource provisioning and low time, memory and other parameters in cloud computing. Further improvement of our approach is to extend this approach for multi server with multi decision of resource provisioning in cloud computing.

VII. REFERENCES

[1]. Sivadon Chaisiri, Bu-Sung Lee, and Dusit Niyato, "Optimal Virtual Machine Placement across Multiple Cloud Providers", 978-1-4244-5336-8/09/$26.00_c 2009 IEEE.

[2]. I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared", in *Proceedings of Grid Computing Environments Workshop (GCE)*, 2008.

[3]. L. M. Vaquero, L. R. Merino, J. Caceres, and M. Lindner, "A Break in the Clouds: Towards a Cloud Definition", *ACM SIGCOMM Computer Communication Review*, pp. 50-55, January 2009.

[4]. P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization", in *Proceedings of ACM Symposium on Operating Systems Principles (SOSP)*, 2003.

[5]. Sivadon Chaisiri, "Optimization of Resource Provisioning Cost in Cloud Computing", IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 5, NO. 2, APRIL-JUNE 2012.

[6]. A.J. Conejo, E. Castillo, and R. Garcı´a-Bertrand, "Linear Programming: Complicating Variables," Decomposition Techniques in Mathematical Programming, chapter 3, pp. 107-139, Springer, 2006.

[7]. J. Linderoth, A. Shapiro, and S. Wright, "The Empirical Behavior of Sampling Methods for Stochastic Programming," Ann. Operational Research, vol. 142, no. 1, pp. 215-241, 2006.

[8]. G. Juve and E. Deelman, "Resource Provisioning Options for Large-Scale Scientific Workflows," Proc. IEEE Fourth Int'l Conf. e-Science, 2008.

[9]. Z. Huang, C. He, and J. Wu, "On-Demand Service in Grid: architecture Design, and Implementation," Proc. 11th Int'l Conf. Parallel and Distributed Systems (ICPADS '05), 2005. [10] Y. Jie, Q. Jie, and L. Ying, "A Profile-Based Approach to Just-in-Time Scalability for Cloud Applications," Proc. IEEE Int'l Conf. Cloud Computing (CLOUD '09), 2009.

[10]. Y. Kee and C. Kesselman, "Grid Resource Abstraction, Virtualization, and Provisioning for Time-Target Applications," Proc. IEEE Int'l Symp. Cluster Computing and the Grid, 2008.

[11]. A. Filali, A.S. Hafid, and M. Gendreau, "Adaptive Resources Provisioning for Grid Applications and Services," Proc. IEEE Int'l Conf. Comm., 2008.

[12]. D. Kusic and N. Kandasamy, "Risk-Aware Limited Lookahead Control for Dynamic Resource Provisioning in Enterprise Computing Systems," Proc. IEEE Int'l Conf. Autonomic Computing, 2006.

[13]. K. Miyashita, K. Masuda, and F. Higashitani, "Coordinating Service Allocation through Flexible Reservation," IEEE Trans. Services Computing, vol. 1, no. 2, pp. 117-128, Apr.-June 2008.

[14]. J. Chen, G. Soundararajan, and C. Amza, "Autonomic Provisioning of Backend Databases in Dynamic Content Web Servers," Proc. IEEE Int'l Conf. Autonomic Computing, 2006.

[15]. L. Grit, D. Irwin, A. Yumerefendi, and J. Chase, "Virtual Machine Hosting for Networked Clusters: Building the Foundations for Autonomic Orchestration," Proc. IEEE Int'l Workshop Virtualization Technology in Distributed Computing, 2006.

[16]. H.N. Van, F.D. Tran, and J.-M. Menaud, "SLA-Aware Virtual Resource Management for Cloud Infrastructures," Proc. IEEE Ninth Int'l Conf. Computer and Information Technology, 2009.

[17]. Junwei Cao, Kai Hwang, "Optimal Multiserver Configuration for Profit Maximization in Cloud Computing", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. XX, NO. YY, MONTH 2012.

[18]. F. I. Popovici and J. Wilkes, "Profitable services in an uncertain world," *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*, 2005.

[19]. J. Sherwani, N. Ali, N. Lotia, Z. Hayat, and R. Buyya, "Libra: a computational economy-based job scheduling system for clusters," *Software – Practice and Experience*, vol. 34, pp. 573-590, 2004.

[20]. C. S. Yeo and R. Buyya, "A taxonomy of market-based resource management systems for utility-driven cluster computing", *Software – Practice and Experience*, vol. 36, pp. 1381-1419, 2006.

[21]. B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and practical limits of dynamic voltage scaling," *Proceedings of the 41st Design Automation Conference*, pp. 868-873, 2004.

[22]. D. Durkee, "Why cloud computing will never be free," *Communications of the ACM*, vol. 53, no. 5, pp. 62-69, 2010.

[23]. R. Ghosh, K. S. Trivedi, V. K. Naik, and D. S. Kim, "End-to-end perform ability analysis for infrastructure-as-a-service cloud: an interacting stochastic models approach," *Proceedings of 16th IEEE Pacific Rim International Symposium on Dependable Computing*, pp. 125-132, 2010.

[24]. K. Hwang, G. C. Fox, and J. J. Dongarra, *Distributed and Cloud Computing*, Morgan Kaufmann, 2012.