ISSN:0975-9646
Sasmita Parida et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 8303-8308

Emperical Resource Allocation Using Dynamic Distributed Allocation Policy in Cloud Computing

Sasmita Parida, Suvendu Chandan Nayak

Department of Computer Science & Engineering
C.V.Raman college of Engineering, Bhubaneswar

Abstract-: **Now a day's Cloud computing is a most attractive computing model of internet technology, which provides resources on-demand as per use and pay basics. Cloud services providers charge for the resource use. So it is important to distribute resources in an efficient way for on demand application of the clients. For resource allocation different parameters are consider like CPU, Main memory, Storage area, Bandwidth, I/O devices et al. By considering these parameters many researchers have proposed different resource allocation algorithm for resource allocation in cloud computing environment. Among these algorithms or policies Dynamic Distributed Allocation Policy (DDAP 1 and DDAP2) is the simplest resource allocation algorithm that is used for CPU allocation. In this work we have studied both DDAP1 and DDAP2 by considering memory as one of the parameter. Basically DDAP2 is not providing a better allocation for memory. We modified some of the policies of DDAP2 for memory allocation, which gives better allocation of memory in cloud computing environment. The memory allocation is improved using DDAP2 by adapting a new policy based on CPU allocation using DDAP2.**

Keywords-: **Pay and Use , resource allocation, DDAP2, on demand, VM, cost.**

I. INTRODUCTION

Now a days cloud computing is so popular because of its several common properties like elasticity and scalability, multi-tenancy, self-managed function capabilities, service billing and metering functions, connectivity interfaces. It has become a challenging technology area, and most of the experts expect that cloud computing will remodel information technology in such a way that an user can fulfill all its demands. Each physical machine has its own resources with fixed amount like CPU, Main memory, storage area, I/O and bandwidth. When a request is send by the user to the datacenter. The virtual machine (VM) will be created according to the request at the physical machine. Due to these platforms the cloud computing technology includes more cost savings, high availability of resource, and easy scalability for user demands.

According to Hurwitz [1] cloud computing is most popular technology which provides a cost effective model and flexible means through which scalable computing power and diverse services (computer hardware and software resources, networks and computing infrastructures), diverse application services, business processes to personal intelligence and collaboration are delivered as services to large-scale global users whenever and wherever they need. For this platform a sole technology is not enough rather mixture of technologies existing before, like: grid computing, utility computing, virtualization or autonomic computing [2]. The main challenge is creation of number of virtual machine in a physical machine to process the user request. The user is unknown about the physical machine. The user request which is called lease consists of different parameters like CPU Time, Main memory, Storage area. To schedule resources different models and algorithms are proposed. Basically the resource allocation is done in terms of leases. Virtualization is the one of important technology that enables virtualization of resources such as storage, processing and network bandwidth. It also describes on-demand Internet applications running as services on that infrastructure. It supports large scale user accesses at distributed locations over the Internet, offers on-demand application services at anytime, and provides both virtual and/or physical appliances for customers.

In cloud computing elasticity is the major characteristics .Which has the ability to create a variable number of virtual machine instances depending on the applications on demands [3,4].In cloud computing the application that are provided to the user according to the user's demand is called Software-as-a-Service (SaaS). Basically this type of service providers are called SaaS providers have the ability to dynamically increase and decrease. This ability is based on use and pay basics of resources whenever an user need. It is an attractive capability which should be done correctly in time by which a cost effective and less expensive as compared to regular traditional hosting [3].

Most of the time it is too difficult for cloud providers to successfully execute all the users' requests which come to them in a instant due to lack of resource or scheduling of resource. As reported in [5] resource allocation model is categories into three ways.

- Effort with a focus on Data Center processing Resources.
- Efforts with a focus on Data center Network Resources.
- Efforts with a focus on Energy Efficient Data Center Resource Allocation.

In this paper we are considering on the Datacenter processing resources. In a datacenter resources can be allocated as static allocation or dynamic allocation. In case of static allocation the resources are allocated before submitting the application to the cloud. The user takes an agreement maximum utilization of resource from the cloud service provider. In this case the resource allocation does not depend upon the resource required for the application. So maximum number of resources cannot be utilized properly. Basically cloud providers follow "Just-in-time" scalability (Dynamic allocation) which is a very challenging concept in cloud computing .The basic idea is releasing and acquiring resources dynamically only when necessary. In traditional approach scaling of the applications are based on the number of users. So there is a need of a true elastic architecture to charge SaaS providers the actual resource usage in "just in time" [6].

To achieve "just in time" in cloud computing environment a cost effective resource allocation policy is required. Resource in cloud computing does not mean only the CPU. Memory, bandwidth, storage area and I/O devices are also in consideration. In this paper the CPU and memory is allocated to the VMs according to their demands. DDAP1 and DDAP2 are basically better for CPU allocation. Instead of proposing another algorithm for memory allocation, we tried to change the policy of DDAP2 for memory allocation by which it can allocate both CPU and memory in cloud computing. The paper allocates memory using DDAP2 by adding new policy to it, which provides better memory allocation. The CPU allocation using DDAP2 is used for the memory allocation, means that CPU allocation is done by DDAP2 [7], then by looking the CPU allocation DDAP2 policy is changed for better memory allocation. Here we are considering the SaaS model of cloud computing.

The rest of the paper is organized as follows: In Section II, the related works has been presented in the area of resource allocation in cloud computing. The CPU and memory allocation using DDAP1 and DDPA2 along with some new policy is described for DDAP2 in Section III. In section IV the result is analyzed which specifies that by changing some policy with DDAP2 we can achieve better memory allocation. However the conclusion and possibilities of the future works are illustrated in Section V.

II. RELATED WORK

A virtual machine is a software computer that, like a physical computer, runs an operating system and applications. The virtual machine is comprised of a set of specification and configuration files and is backed by the physical resources of a host. Every virtual machine has virtual devices that provide the same functionality as physical hardware, and have additional benefits in terms of portability, manageability and security. Virtual machines are the key component in a virtual infrastructure. Every virtual machine has CPU, memory and disk as resources. A virtual machine is associated to a particular datacenter, host, cluster, or resource pool. Basically in cloud computing the virtual machines are associated with datacenter.

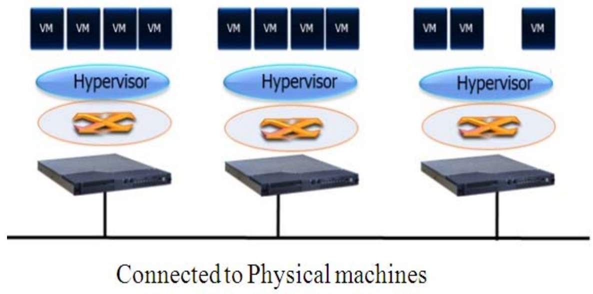

Figure 1: VMs Created in Physical Machines

In Fig 1 number of virtual machines can be deployed to the datacenters. Every virtual machine has virtual devices that provide the same function as physical hardware. When a virtual machine is no longer needed, it can be removed from the datacenters, like wise whenever it is required it can be created. It is very important task to allocate resources to the virtual machine. Number of research papers has been published for resource allocation for cloud environment within last few years.

Current cloud virtualization mechanisms do not provide cost-effective pay-per-use model for Software-as-a- Service (SaaS) applications [8]. Recently a market oriented resource allocation scheme which is integrated along with both customer-driven service management and computational risk management to sustain service level agreement (SLA) is proposed by Rajkumar Buyya et al.[9]. Fang Y et al. [10] proposed a two-level task-scheduling mechanism which is based on load balancing of resources. Users' applications are assigned to VMs in the first-level scheduling by creating a task description for each VM. Then, by applying certain rules proper host resources are allocated to VMs in the second level. However, this scheduling algorithm quite complex due to the presence of two levels of scheduling. It is a big challenge for efficient resource scheduling algorithm design and implementation (since general scheduling problem is NP complete) [11].

A net work aware resource allocation in distributed clouds is proposed [12].The authors minimized the maximum distance, or latency, between the selected data centers. Where the user has large task and it needs more than one virtual machine. Haizea also provides a better advanced reservation and deadline sensitive type of leases. Capacity leasing in cloud systems using the open nebula engine is discussed in [13]. In [14] more on resource leasing and the art of suspending virtual machines is briefly discussed. Sotomayor also discussed combining batch execution and leasing using virtual machines [15].Basically a virtual machine consists of CPU, memory and disk space as resources. CPU virtualization gives importance on performance and runs directly on the virtual machine processor. These physical resources are used when possible. The virtualization layer in cloud computing runs only when the instructions are required to make virtual machines operate .We can add or configure different virtual machine properties at the time of virtual machine creation in a physical machine.

Virtual environments can vastly improve resource utilization, but optimizing that utilization will require careful management of VM resources using capable tools that can help administrators watch demands, spot trends over time and allocate new resources with minimal disruption to each workload. Virtual machine monitors (VMMs) like Xen provide a mechanism for mapping virtual machines (VMs) to physical resources [16]. VM live migration technology makes it possible to change the mapping between VMs and PMs while applications are running [17].

The challenges of SaaS applications for application vendors and providers are discussed, taking into account the need for customization of SaaS applications [18]. Basically in cloud computing tenant based resource allocation method is used. That solution was introduced and tested with regard to CPU and memory utilization [19]. Li et.al[20] proposed a non-preemptive priority M/G/1 queuing model for task-scheduling for on demand users in cloud computing. The tasks are scheduled only by their execution time. In this paper we allocate the resource both CPU and memory using DDAP2 [7] to the VMs, according to their demands. A tenant is a customer that uses or provides a SaaS application. For scalability cloud computing provider offers one application to the multiple tenants, so SaaS application must be multitenant aware [21].In this paper we focused on CPU and memory allocation using DDAP1 and DDAP2. DDAP2 is very simple method for CPU allocation by using the share ration. By applying share ration concept in DDAP2 for memory allocation it is not providing proper memory allocation. The result is discussed in section-IV by the graph. Whenever we changed the policy of DDAP2 a better memory allocation is found out.

III. DYNAMIC DISTRIBUTED ALLOCATION POLICY (DDAP)

The concept of resource allocation to VM is very challenging one in cloud computing. In cloud computing resources allocated to each VM must be tailored to the demands of each workload. It is more challenging to find such a concept whose goal to "right size" the computing resources allocated to each individual VM. So that each VM has enough resources to handle peak resource demands while still sustaining an acceptable level of performance.

In cloud computing hypervisor allocates resources automatically when a virtual machine (VM) starts up, the allocation process is rarely tailored for the individual workload. During resource allocation two situations occurs like “under allocated” and “over allocating”. When resources are under-allocated, the workload will run poorly. In case of over allocating resources doesn't harm the VM, but it can waste the very computing resources that you're trying to optimize. For cloud computing, service provider always prefers over-allocated resources for the VM due to increasing demand of cloud computing. The unused resources can be used by other VMs by using different allocation mechanism.

In [7] the author analyzed Static Allocation Policy (SAP), Dynamic Distributed Allocation Policy 1 (DDAP1) and Dynamic Distributed Allocation Policy 2. In DDAP1 sharing policy is used whereas in DDAP2 unmet demands are used for resource allocation. In both the policies the authors had discussed only for the CPU utilization. In this study memory allocation is based on CPU utilization using DDAP2. In the first phase on demand CPU is allocated, then according to the allocated CPU, memory will be allocated in second phase. The study of DDAP2 is organized as 1-:DDAP2 for CPU allocation, 2-:DDAP2 for memory allocation and finally DDAP2 with changed policy for memory allocation.

3.1: DDAP1 for CPU Allocation

Before discussing DDAP1 let us discuss SAP(Static allocation Policy) which is rearly used in cloud computing.The noatations used in the SAP are as:ρ for capacity per share, S for share, and C for total capacity.The allocation algorithm of SAP is as below.

*SAP Algorithm***:**

Step 1:for i=1 to n
Read VM(i) and S(i)
Step 2:Calculate
$\rho = C / \sum S(i)$
Step 3:for i=1 to n allocate each VM as.
$A(i)=\rho * S(i)$

This algorithm is independent of each VMs demand value. So there may be chance of getting high share value to a VM having low demand value.To solve this type of problem dynamic distributed allocation policy(DDAP1 and DDAP2) is used.In DDAP1 for each VM demand is an upper bound limit for the allocation of the capacity.

DDAP1 Algorithm:

Step 1:Repeat the following steps for each VM if $C <= \sum d(i)$ where d for demand value of each VM.
Step 2: for i=1 to n
Allocate to each VM by considering capacity per share parameter ρ^* such that:
$A(i)=\min\{ d(i), \rho^* * S(i) \}$
and
$\sum A(i)=C$

DDAP2 Algorithm:

Here each VM(i) contain two parameters for its allocation as:
λ (i) = Lower bound allocation
μ (i)= Upper bound allocation
Step 1: for i=1 to n
Allocate each VM with its lower bound allocation λ (i).
Step 2:Claculate unmeet need of VM(i)
$\delta(i)=\epsilon / (\mu (i)- \lambda (i))$
$\epsilon \rightarrow$Remaining capacity calculated as
$\epsilon=C-\sum\lambda(i)$
Step 3:for each i=1 to n
do
$A(i)=\lambda(i)+ ((\delta(i)/ \sum\delta(j)) * \epsilon)$

Let take a example where a physical system has 10 GHz of CPU, and with 12GB of memory (RAM). There are 4 number of virtual machines deployed in it having different demands as shown in table 1. Here the CPU and share data is taken from [7] for better understanding.

VM parameters	VM1	VM2	VM3	VM4
CPU in GHz	1	8	1	3
Share	400	400	100	100
Memory in GB	3	7	2	3

TABLE 1: ON DEMAND RESOURCE OF 4 VMS.

So the demanded CPU utilization is 1+8+1+3=13. But the physical system has CPU with maximum 10GHz. In case of cloud computing we can't ignore the demand of the VM according to client's requirements. It can be easily solved by DDAP2. According to DDAP2 allocate the CPU with respect to the share of the VMs. Means that if on demand CPU is less than the share of the VM, allocate the CPU completely. If the demanded CPU of a VM is greater than the share, allocate the CPU according to the share value. Consider the VM1 and VM2 from the table-1.The VM1 has share of 400 and the demanded CPU is 1GHz. So according to the share value VM1 demands less CPU utilization. According to DDAP2 1GHz is allocated to VM1 out of 10GHZ CPU from physical machine. For VM2 the share is 400 and demand is 8 GHz. Here the demanded CPU is more than the share ratio. So we can assign 4GHz according to the share ratio of VM2.The CPU allocation for rest of the VMs is shown in the Table 2.

VM parameters	VM1	VM2	VM3	VM4
CPU in GHz	1	8	1	3
Share	400	400	100	100
Allocated CPU in GHz	1	4+2=6	1	1+1=2
Allocation with %	100%	75%	100%	66%

TABLE 2: ALLOCATION OF CPU USING DDAP2.

Likewise for VM3 we can assign 1GHz where the share is 100. VM4 demands for 3GHz but having share value as 100. So 1GHz is allocated to it. So the total allocation is 1+4+1+1=7GHz of CPU out of 10 GHZ from the physical machine. The amount of wastage of CPU which is unallocated is 10-7=3GHz.The rest of the demand CPU for the VMs is 8-4=4 for VM2 and 3-1=2 for VM4.So to fulfill all the demand of the VMs we need 4+2=6 GHz. But we have only 3GHz as unused. So out of 3GHz VM2 and Vm4 takes 2GHz and 1GHz respectively as shown in the Table 2. Here the percentages of the allocated CPU with respect to the demands of the VMs are calculated. These values will be used for memory allocation in section 3.2.

3.2: DDAP2 for Memory Allocation

Let us apply DDAP2 for the memory allocation as like CPU allocation. The physical system has 12 GB memory and the demands of the memory by the different VMs are shown in Table 1. The allocated demanded memory for the VMs is calculated in similar fashion that is used for CPU allocation in section 3.1. So the memory allocated using DDAP2 is shown in Table 3. This allocation of the memory can be improved for better memory allocation using DDAP2. Further our discussion is how DDAP2 can give a better memory allocation.

VM parameters	VM1	VM2	VM3	VM4
Memory in GB	3	7	2	3
Share	400	400	100	100
Allocated Memory in GB	3	4+1.5=5.5	1+0.5=1.5	1+1=2
Allocation with %	100%	78.57%	75%	66.67%

TABLE 3: ALLOCATION OF MEMORY USING DDAP2

In cloud computing the memory should be allocated in such a way that it can meet all the requirements of the client. The cloud providers should provide the memory allocation mechanism in an efficient way to full fill the demand as far as possible. In the Table 3 we have calculated the percentage of memory allocation with respect to the demand of memory of all VMs. Some of the VMs get 100% according to the demand (VM1) and some of have less percentage of their demand (VM2=78.57%, VM3=75% and VM4=66.67%).

To equalize the allocated on demand memory allocation we changed DDAP2 with same example. For VM2 the demand of the memory is 7 GB but 5.5 GB are allocated. For VM1 both CPU and memory allocation is 100% where as for VM4 it is around 66%. Considering the above case we changed the policy for the same problem for fair memory allocation. Before allocating the memory CPU allocation is consider from the table 2.

DDAP2 Algorithm for memory:

Here each VM(i) contain two parameters for its allocation as:

λ (i) = Lower bound allocation

μ (i)= Upper bound allocation

Step 1: for i=1 to n

Allocate each VM with its lower bound allocation λ (i) and without allocating the VM having A(i)==d(i)

Step 2:Claculate unmeet need of VM(i)

δ(i)=Є / (μ (i)- λ (i))

Є→Remaining capacity calculated as

Є=C-∑λ(i)

Step 3:for each i=1 to n do

A(i)=λ(i)+ ((δ(i)/ ∑δ(j)) * Є)

In the above algorithm we have changed the step 1 where we are considering the CPU allocation of VM before allocating the memory.Considering the share the CPU allocation is done for VM1. The share is 400 and the demand CPU allocation is 1GHz. According to the DDAP2 the demanded 1GHz is allocated because the ratio of share is greater than the demand of CPU. Similarly 3GB memory is allocated to VM1 which did not lead towards a better memory allocation. In the changed mechanism we dropped the memory allocation for VM1, though it has 100% CPU allocation. But VM3 is not dropped even if its CPU allocation is 100% from Table 2. Because the share ratio is 1 and the demanded memory allocation is 2GB. So 1GB memory is allocated to VM3. Likewise 4GB for VM2, 1GB for VM4 and no initial allocation for VM1. The total new allocation of Table 3 is shown in table 4.The total initial memory allocation is: VM1+VM2+VM3+VM4 which is 0+4+1+1= 6GB.The total memory is 12GB.So the rest of the memory that will be allocated is 12-6=6GB. Out of 6GB for VM1 the allocation is 2GB, for VM2 the allocation is 2GB, for VM3 the allocation is 0.67 and for VM4 the allocation is 1.34. The final allocation of the memory is shown in the following table.

VM parameters	VM1	VM2	VM3	VM4
Memory in GB	3	7	2	3
Share	400	400	100	100
Allocated Memory in GB	2	4+2 = 6	1+0.67 = 1.67	1+1.34 = 2.34
Allocation with %	66.7%	85.7%	83.5%	78%

TABLE 4: MEMORY ALLOCATION USING THE PROPOSED MECHANISM. (THE 4 DECIMAL VALUES ARE CONSIDERED AFTER DECIMAL POINT)

The percentage of the memory allocation is calculated, which is used for the result analysis of the memory allocation after changing DDAP2 to VMs according to the demands.

IV: RESULT ANALYSIS

In this section we present the result and analyze the memory allocation using DDAP2 and allocation of memory after changing the policy of DDAP2.The simulation is done by using CLOUDSIM with 4 VMs in windows2007 platform. In the table 3 when we are allocating the memory using DDAP2 VM1 gets 5GB because its share value is 400. So the amount of memory demanded by VM1 is allocated. The allocation percentage of memory for VM1 is 100%. Where VM2 needs 9GB memory and DDAP2 allocates only 7GB.So the percentage of memory allocation is 72% but both VM1 and VM2 have the same share value 400 as shown in table 3. For VM4 the percentage of memory allocation is 66% which is very less. In table 4 we are not allocating the memory to VM1 initially because its CPU utilization is 100%. The table 4 shows the allocation for VM1 is 69% using our modified DDAP2. Other VMs memory allocation percentages are increased which leads towards a better memory allocation.

As compared to DDAP2 our modified mechanism allocated memory in a better way which is nearly to the demand of the clients and maintains nearly equal allocation as shown in the chart in the figure 2. For VM1 the memory allocation percentage is reduced to 69% according to modified DDAP2. But other 3 VMs have got more memory allocation as compared to DDAP2.

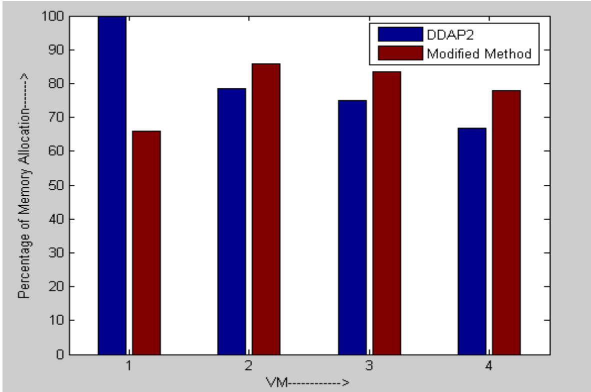

Fig 2: Memory allocation chart.

In the table 3 and 4 the percentages of memory allocation using DDAP2 and modified DDAP2 is shown in different colors for better understanding and analysis. As shown in the chart the modified DDAP2 allocates memory in a better way as compared to DDAP2.

V: CONCLUSION AND FUTURE WORK

Resource allocation to VM is not only CPU. There is also memory, storage space and bandwidths. DDAP2 is better way to allocate CPU among the VMs but not so much good for memory allocation. In this paper we are allocating the CPU and memory at a time to the VMs. In case of memory allocation using DDAP2 we need to observe the percentage of CPU utilization by which the memory can be allocated in a better way. Table 4 shows better memory allocation on the demand of VMs as compared to table 3.

The future work can be carried out for allocation of storage space and bandwidth along with CPU and memory by finding a relationship in between them. By studying the relationship a better resource allocation might be possible for the VMs. The proposed allocation mechanism of the DDAP2 for memory allocation can be implemented and observed in cloud computing for VMs.

REFERENCES

[1] J. Hurwitz, M. Kaufman, and R. Bloor, “Cloud Computing for Dummies,” Wiley Publishing, Inc. 2010.

[2] Zhang, Q. et al. 2010. Cloud computing: State-of-the-art and research challenges. Journal of Internet Services and Applications. 1, 1 (2010), 7–18.

[3] M. Armbrust, et al. Above the clouds: a Berkeley view of cloud computing, electrical engineering and computer sciences, Technical Report No. UCB/EECS-2009-28, University of California at Berkeley, February 2009.

[4] R. Buyya, C. Shin Yeo, S. Venugopal, J. Broberg, Brandic, Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems 25 (6) (2009) 599–616.

[5] Mohamed Abu Sharkh, Abdelkader Ouda, and Abdallah Shami,”A Resource Scheduling Model for Cloud Computing Data centers,”IEEE Communication Magazine,2013.

[6] J. Wong, G. Iszlai, M.L. Ye Hu, Resource provisioning for cloud computing, in: Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research, November 2009.

[7] G Shanmuganathan, Ajay Gulati,Peter Vaman ,Defragmenting the cloud Using Demand-based resource allocation,SIGMETRIC,13 Proceedins of ACM SIGMETRIC international conference measurement and modeling of computer systems,Volume 4 ,Issue 1,2013.

[8] Y. Jie, Q. Jie, L. Ying, A profile-based approach to just-in-time scalability for cloud applications, in: CLOUD'09, IEEE International Conference on Cloud Computing, September 2009.

[9] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic, Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility, Future Generation Computer Systems, Volume 25,Number 6, Pages: 599-616, ISSN: 0167-739X, Elsevier Science,Amsterdam, The Netherlands, June 2009.

[10] Fang Y, Wang F, Ge J. A task scheduling algorithm based on load balancing in cloud computing. Web Information Systems and Mining, Lecture Notes in Computer Science 2010; 6318: 271–277.

[11] Weiwei Lin, Baoyun Peng, Chen Liang ,etc"Novel Resource Allocation Model and Algorithms for Cloud Computing" IEEE 2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies,2013.

[12] Mansoor Alicherry, T.V. Lakshman" Net work Aware Resource Allocation in Distributed Clouds" 2012 Proceedings IEEE INFOCOM, 2012.

[13] Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I.: Capacity leasing in cloud systems using the opennebula engine. In: Cloud Computing and Applications, CCA 2008 (2009).

[14] Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I.: Resource leasing and the art of suspending virtual machines. In: 11th IEEE International Conference on High Performance Computing and Communications, pp. 59–68. IEEE (2009).

[15] Sotomayor, B., Keahey, K., Foster, I.: Combining batch execution and leasing using virtual machines. In: Proceedings of the 17th International Symposium on High Performance Distributed Computing, HPDC 2008, pp. 87–96. ACM (2008)

[16] 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 Proc. of the ACM Symposium on Operating Systems Principles (SOSP'03), Oct. 2003.

[17] M. Nelson, B.-H. Lim, and G. Hutchins, "Fast transparent migration for virtual machines," in Proc. of the USENIX Annual Technical Conference, 2005.

[18] R. Mietzner, A. Metzger, F. Leymann, K. Pohl, Variability modeling to support customization and deployment of multi-tenant-aware software as a service applications, in: PESOS '09: Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems, May 2009.

[19] Espadas, J. et al. 2011. A tenant-based resource allocation model for scaling Software-as-a-Service applications over cloud computing infrastructures. (2011).

[20] Luqun L. An optimistic differentiated service job scheduling system for cloud computing service users and providers.Proceedings of the third International Conference on Multimedia and Ubiquitous Engineering, 2009, 295–299.

[21] R. Mietzner, F. Leymann, M.P. Papazoglou, Defining composite configurable saas application packages using SCA, variability descriptors and multi-tenancy patterns, in: ICIW '08, Third International Conference on Internet and Web Applications and Services, June 2008.