International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 6 (2018) pp. 4568-4575


# Cloud Environment: A Review on Dynamic Resource Allocation Schemes

**P. Prathap Nayudu[a], K. Raja Sekhar[b]**

*[a]Research Scholar, K L University, Vaddeswaram, Guntur, Andhra Pradesh, India.*
*[b]Professor, Dept. of CSE, K L University, Vaddeswaram, Guntur, Andhra Pradesh, India.*
*Corresponding author*

**Abstract**

Cloud computing environment provisions the supply of computing resources on the basis of demand, as and when needed. It builds upon the advances of virtualization and distributed computing to support cost efficient usage of computing resources, emphasizing on resource scalability and on-demand services. It allows business outcomes to scale up and down their resources based on needs. Managing the customer demand creates the challenges of on demand resource allocation. Further, they can make use of company-wide access to applications, based on a pay-as-you-go model. Hence there is no need for getting licenses for individual products. Virtual Machine (VM) technology has been employed for resource provisioning. It is expected that using virtualized environment will reduce the average job response time as well as executes the task according to the availability of resources. Effective and dynamic utilization of the resources in cloud can help to balance the load and avoid situations like slow run of systems. In this paper, various resource allocation strategies and their challenges are discussed in detail. It is believed that this paper would benefit both cloud users and researchers in overcoming the challenges faced.

**Keywords** - Cloud Computing, Dynamic Resource Allocation, Resource Management, Resource Scheduling.

## INTRODUCTION

Cloud is a type of parallel and distributed system consisting of a collection of virtualized and interconnected computers that are dynamically provisioned and presented as one or more unified computing resource based on Service Level Agreements (SLA) established through negotiation between the service provider and consumers. Cloud computing is an internet -based computing in which large groups of remote servers are networked to allow sharing of data-processing tasks, centralized data storage, and an online access to computer services or resources. It relies on the sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. Cloud computing also focuses on maximizing the effectiveness of the shared resources. Cloud resources are not only shared by multiple users but are also dynamically re-allocated on demand. The main enabling technology is virtualization. Virtualization software allows a physical computing device to be electronically separated into one or more "virtual" devices, each of which can be easily used and managed to compute tasks. Virtualization provides the agility required to speed up IT operations, and reduces costs by increasing infrastructure utilization.

Scheduling is important for an operating system. CPU scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated CPU time. When a job is submitted to a resource manager, the job waits in a queue until it is scheduled and executed. The time spent in the queue, or wait time, depends on several factors, including job priority, the load on the system, and availability of requested resources. Turnaround time represents the elapsed time between when the job is submitted and when the job is completed. It includes the wait time as well as the jobs actual execution time. Response time represents how fast a user receives a response from the system after the job is submitted. Resource utilization during the lifetime of the job represents the actual useful work that has been performed. System throughput is defined as the number of jobs completed per unit time. Mean response time is an important performance metric for users, who expect minimal response time.

In a typical production environment, many different jobs are submitted to the cloud. So, the job scheduler software must have interfaces to define workflows and/or job dependencies, execute the submitted jobs automatically. The cloud broker has pre-configured and stored in the cloud all the necessary VM images to run users' jobs. All the incoming jobs are enquired into a queue. A system-level scheduler, running on a dedicated system, manages all the jobs and a pool of machines, and decides whether to provision a new VM from clouds and/or to allocate jobs to VMs. The scheduler is executed periodically. At each moment, the scheduler performs five tasks: (1) Predicting future incoming workloads; (2) Provisioning necessary VMs in advance, from clouds; (3) Allocating jobs to VM; (4) Releasing idle VMs if its Billing Time Unit (BTU) is close to increase; (5) If the time of un-allocated jobs is high, starting the necessary number of VMs.

Cloud computing builds upon the advances of virtualization and distributed computing to support cost efficient usage of computing resources, emphasizing on resource scalability and on -demand services. Cloud computing allows business outcomes to scale up and down their resources based on needs. Managing the needs of the customer creates the challenges of on -demand resource allocation. Virtual machine technology has been applied for resource provisioning. Hence VM are allocated to the user based on characteristics of the job. Low priority jobs should not delay the execution of high priority jobs. This scenario leads to resource contention between low and high priority jobs to access resources. The outcome of the paper is priority-based

pre-emption policy that improves resource utilization in a virtualized environment.

The remainder of this paper has been organized as follows. Section 2 gives a brief review of related works regarding resource allocation in a cloud environment. Section 3 gives different resource allocation strategies at a glance, and finally Section 4 concludes the paper.

## RELATED WORK

IaaS cloud allocates resources to competing requests based on pre-defined resource allocation policies. Presently, most of the cloud providers rely on simple resource allocation policies like immediate and best effort. Amazon EC2 is a public cloud which provides computing resources for the general public on pay-per-use model. Eucalyptus and Open Nebula are cloud toolkits which can be used to setup a cloud on local infrastructure.

Haizea is an open source resource lease manager that can be used as a scheduler for Open Nebula and Haizea provides the only Virtual Infrastructure (VI) management solution offering advanced reservation of capacity and configurable VM placement policy. Sometimes it is not possible for cloud providers to satisfy all the requests which come to them on immediate basis due to lack of resources [7]. Haizea tries to address this issue. User requests computational resources from Haizea in the form of lease. The lease is accepted by Haizea if and only if Haizea can assure the resource allocation policy requested by this lease. Assuring the resource allocation policy means providing requested resources for requested duration at a requested start time. Haizea will then reserve the resources for this lease. Whenever start time of these reservation comes, Haizea allocates resources in the form of VMs. Haizea assumes the best-effort leases are preemptable and they do not have any time constraints. Immediate and advance reservation leases are non-preemptable and have time constraints. It will preempt best -effort leases whenever the resources are required for advance reservation or immediate leases. There is no guarantee that a submitted best-effort lease will get resources for completion within a certain time limit. If the system is flooded with lots of advances and immediate leases, then best – effort leases will not have enough resources to run on. The consumers of best-effort lease may not like to wait as long to get resources. They will start submitting their requests as advance reservation leases rather than best-effort leases, to be assured that the submitted requests will be completed within a certain time limit. As there are very less best-effort leases, the system utilization will go down. To handle this situation, deadlines are associated with best-effort leases. These kinds of leases are called deadline sensitive leases. They are assumed to be preemptable. It is preemptable only if the scheduling algorithm of Haizea can assure that it can be completed before its deadline. Thus, it will assure the consumers that their request will be completed within a certain time limit.

VM-based resource reservation (i.e.) the reservations of CPU, memory and network resources for individual VM instances, as well as for VM cluster. The fundamental goal is to enable an application to request the creation of virtual machines and clusters based on high -level specifications of both the VMs' environments and its desired QoS [1]. A model for predicting various run-time overheads involved in using virtual machines, allowing us to efficiently support advance reservations [2]. An approach that uses leasing, as the fundamental resource provisioning abstraction for both the best effort and advance reservation requests. The job abstraction used by batch scheduler ties together the provisioning of resources for the job and its execution, with resource provisioning typically happening as a side-effect of job submission [3]. The Software as a Service (SaaS) provider leases resources, from cloud providers and also leases software as services to SaaS users. The SaaS providers aim at minimizing the payment of using VMs from cloud providers, and want to maximize the profit earned through serving the SaaS user's requests [4]. To reduce the impact of pre-empting VM, policies that determine the proper set of lease(s) for pre-emption are proposed [5].

Batch schedulers implement the backfilling algorithm, but with different variants. A well-known variant is Conservative backfilling where a job enters the waiting queue with an associated start time when a job is submitted to the scheduler. Some jobs in the queue can then be reordered with an earlier start time if they do not delay the already allocated jobs. A variation of this backfilling is aggressive backfilling where the scheduler attributes a start time for the first job in the queue and all the other jobs in the queue can be reorganized at any time if they do not delay the start time of the first job. Haizea comes with backfilling as one of its default scheduling actions. Virtualization allows creating additional virtual processors on physical ones to reduce the problem of scheduling both sequential and parallel jobs. The researchers use virtualization of cloud nodes to manage the time spent by all running tasks on each processor and share them with other tasks. Time sharing between users is however not always realistic on cloud as the applications are often tuned to get the best performance with the assumption that they run alone on one processor.

### Resource Allocation Strategies at Glance

The input parameters to RAS and the way of resource allocation vary based on the services, infrastructure and the nature of applications which demand resources. The schematic diagram in Fig.1 depicts the classification of Resource Allocation Strategies (RAS) proposed in cloud paradigm. The following section discusses the RAS employed in the cloud.

### Execution Time

Different kinds of resource allocation mechanisms are proposed in the cloud. In the work by Jiani at.al [8], actual task execution time and pre-emptible scheduling is considered for resource allocation. It overcomes the problem of resource contention and increases the resource utilization by using different modes of renting computing capacities. But estimating the execution time for a job is a hard task for a user

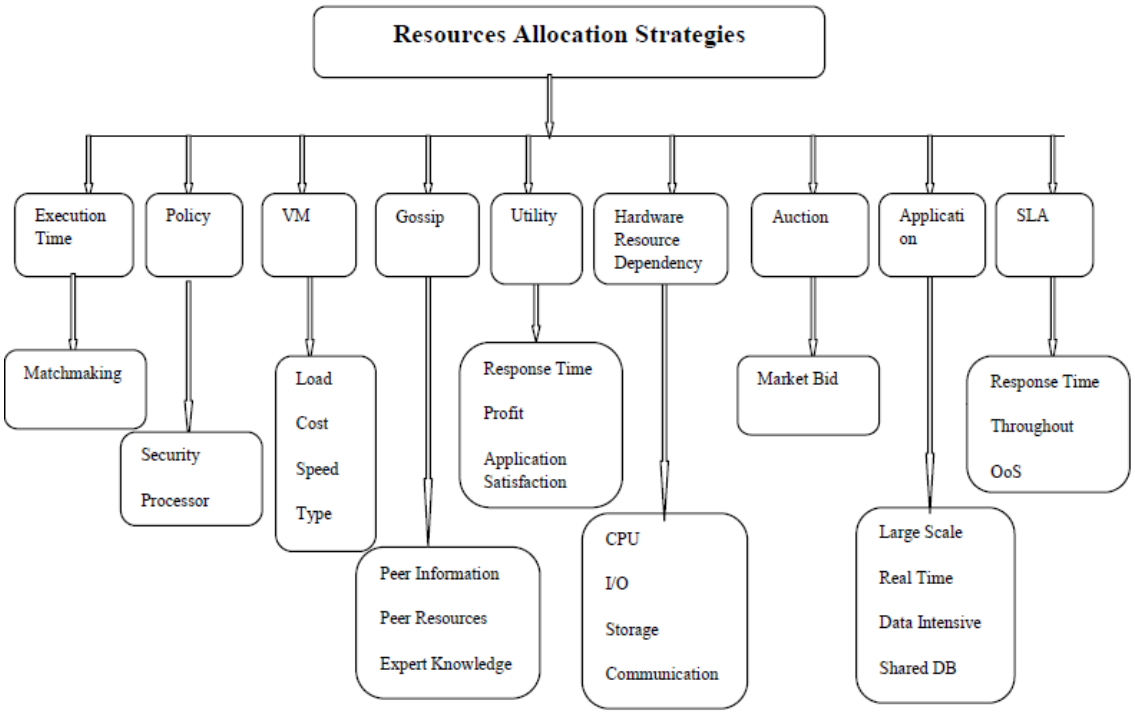


**Figure 1.** Resource Allocation Strategies in Cloud Computing

and errors are made very often. But the VM model considered in [8] is heterogeneous and proposed for IaaS.

Using the above-mentioned strategy, a resource allocation strategy for distributed environment is proposed by Jose et al. [9]. Proposed matchmaking (assign a resource to a job) strategy in [9] is based on Any-Schedulability criteria for assigning jobs to opaque resources in a heterogeneous environment. This work does not use detailed knowledge of the scheduling policies used at resources and subjected to AR's (Advance Reservation).

**Policy**

Since centralized user and resource management lacks in scalable management of users, resources and organization-level security policy, Dongwan et al. [10] has proposed a decentralized user and virtualized resource management for IaaS by adding a new layer called domain in between the user and the virtualized resources. Based on role based access control (RBAC), virtualized resources are allocated to users through domain layer.

One of the resource allocation challenges of resource fragmentation in multi-cluster environment is controlled by the work given by Kuo-Chan et al. [11], which used the most-fit processor policy for resource allocation. The most-fit policy allocates a job to the cluster, which produces a leftover processor distribution, leading to the number of immediate subsequent job allocations.

It requires a complex searching process, involving simulated allocation activities, to determine the target cluster. The clusters are assumed to be homogeneous and geographically distributed. The number of processors in each cluster is binary compatible. Job migration is required when load sharing activities occur.

Experimental results shows that the most-fit policy has higher time complexities, but the time overheads are negligible compared to the system long time operation. This policy is practical to use in a real system.

**Virtual Machine (VM)**

A system which can automatically scale its infrastructure resources is designed in [12]. The system composed of a virtual network of virtual machines capable of live migration across multi- domain physical infrastructure. By using the dynamic availability of infrastructure resources and dynamic application demand, a virtual computation environment is able to automatically relocate itself across the infrastructure and scale its resources. But the above work considers only the non-preemptable scheduling policy. Several researchers have developed efficient resource allocations for real time tasks on multiprocessor systems. But the studies, scheduled tasks on fixed number of processors. Hence it lacks in scalability feature of cloud computing [13]. Recent studies on allocating cloud VMs for real time tasks focus on different aspects like infrastructures to enable real-time tasks on VMs and selection of VMs for power management in the data centre. But the work by Karthik et al. [13], have allocated the resources based on the speed and cost of different VMs in IaaS. It differs from other related works, by allowing the user to select VMs and reduces cost for the user. Users can set up and boot the required resources and they have to pay only for the required resources. It is

implemented by enabling the users to dynamically add and/or delete one or more instances of the resources on the basis of VM load and the conditions specified by the user. The above mentioned RAS on IaaS differs from RAS on SaaS in cloud because SaaS delivers only the application to the cloud user over the internet.

Zhen Kong et al. have discussed mechanism design to allocate virtualized resources among selfish VMs in a non - cooperative cloud environment in [14]. By non-cooperative means, VMs care essentially about their own benefits without any consideration for others. They have utilized stochastic approximation approach to model and analyze the QoS performance under various virtual resource allocations. The proposed stochastic resource allocation and management approaches enforced the VMs to report their types truthfully and the virtual resources can be allocated efficiently. The proposed method is very complex and it is not implemented in a practical virtualization cloud system with real workload.

**Gossip**

Cloud environment differs in terms of clusters, servers, nodes, their locality reference and capacity. The problem of resource management for a large-scale cloud environment (ranging to above 100,000 servers) is addressed and general Gossip protocol is proposed for fair allocation of CPU resources to clients.

A gossip-based protocol for resource allocation in large-scale cloud environments is proposed in [15]. It performs a key function within distributed middleware architecture for large clouds. In the paper, the system is modeled as a dynamic set of nodes that represent the machines of cloud environment. Each node has a specific CPU capacity and memory capacity. The protocol implements a distributed scheme that allocates cloud resources to a set of applications that have time-dependent memory demands and it dynamically maximizes a global cloud utility function. The simulation results show that the protocol produces optimal allocation when memory demand is smaller than the available memory in the cloud and the quality of the allocation does not change with the number of applications and the number of machines. But this work requires additional functionalities to make resource allocation scheme is robust to machine failure, which spans several clusters and data centers.

But in the work, cloud resources are being allocated by obtaining resources from remote nodes when there is a change in user demand and have addressed three different policies to avoid over-provisioning and under-provisioning of resources. Recent research on sky computing focuses on bridging multiple cloud providers using the resources as a single entity which would allow elastic site for leveraging resources from multiple cloud providers. Related work is proposed in [12] but it is considered only for pre-emptible tasks. Yang et al. [16] have proposed a profile based approach for scaling the applications automatically by capturing the experts' knowledge of scaling application servers as a profile. This approach greatly improves the system performance and resource utilization.

In paper [8], Gossip based co-operative VM management with VM allocation and cost management is introduced. By this method, the organizations can cooperate to share the available resources to reduce the cost. Here the cloud environments of public and private clouds are considered. They have formulated an optimization model to obtain the optimal virtual machine allocation. The network game approach is adopted for the cooperative formation of organizations so that none of the organizations want to deviate. This system does not consider the dynamic co-operative formation of organizations.

**Utility Function**

There are many proposals that dynamically manage VMs in IaaS by optimizing some objective function such as minimizing the cost function, cost performance function and meeting QoS objectives. The objective function is defined as Utility property which is selected based on measures of response time, number of QoS, targets met and profit etc.

There are few works [17] that dynamically allocate CPU resources to meet QoS objectives by first allocating requests to high priority applications. The authors of the papers do not try to maximize the objectives. Hence the authors' Dorian et al. proposed Utility (profit) based resource allocation for VMs which use live VM migration (one physical machine to other) as a resource allocation mechanism [18]. This controls the cost-performance trade-off by changing VM utilities or node costs. This work mainly focuses on scaling CPU resources in IaaS. A few works are also there that use live migration as a resource provisioning mechanism, but all of them use policy based heuristic algorithm to live migrate VM which is difficult in the presence of conflicting goals.

For multitier cloud computing systems (heterogeneous servers), resource allocation based on response time as a measure of the utility function is proposed by considering CPU, memory and communication resources in [19]. HadiGoudarzi et al. characterized the servers based on their capacity of processing powers, memory usage and communication bandwidth.

For each tier, the requests of the application are distributed among some of the available servers. Each available server is assigned to exactly one of these applications tiers i.e. server can only serve the requests on that specified server. Each client request is dispatched to the server using queuing theory and this system meets the requirement of SLA such as response time and utility function based on its response time. It follows the heuristics called force-directed resource management for resource consolidation. But this system is acceptable only as long as the client behaviors remain stationary. But the work proposed in [20] considers the utility function as a measure of application satisfaction for specific resource allocation (CPU, RAM). The system of data centre with single cluster is considered in [20] that support heterogeneous applications and workloads, including both enterprise online applications and CPU- intensive applications. The utility goal is computed by Local Decision Module (LDM) by taking current work load of the system. The LDMs interact with Global Decision Module (GDM) and

that is the decision making entity within the autonomic control loop. This system relies on a two-tier architecture and resource arbitration process that can be controlled through each application's weight and other factors.

***Hardware Resource Dependency***

In paper [21], to improve the hardware utilization, Multiple Job Optimization (MJO) scheduler is proposed. Jobs could be classified based on hardware-resource dependency such as CPU-bound, Network I/O-bound, Disk I/O bound and memory bound. MJO scheduler can detect the type of jobs and parallel jobs of different categories. Based on the categories, resources are allocated. This system focuses only on CPU and I/O resource.

Eucalyptus, Open Nebula and Nimbus are typical open source frame works for resource virtualization management [22]. The common feature of these frameworks is to allocate virtual resources based on the available physical resources, expecting to form a virtualization resource pool decoupled with physical infrastructure. Because of the complexity of virtualization technology, all these frameworks cannot support all the application modes. The system, called Vega Ling Cloud proposed in paper [22] supports both virtual and physical resources leasing from a single point to support heterogeneous application modes on shared infrastructure.

Cloud infrastructure refers to the physical and organizational structure needed for the operation of cloud. Many recent researches address the resource allocation strategies for different cloud environment. Xiaoping Wang et al. have discussed adaptive resource co- allocation approach based on the CPU consumption amount in [23]. The stepwise resource co-allocation is done in three phases. The first phase determines the co-allocation scheme by considering the CPU consumption amount for each physical machine (PM). The second phase determines whether to put applications on PM or not by using a simulated annealing algorithm which tries to perturb the configuration solution by randomly changing one element. During phase 3, the exact CPU share that each VM occupies is determined and it is optimized by the gradient climbing approach. This system mainly focuses on CPU and memory resources for co-allocation and does not consider the dynamic nature of resource request.

HadiGoudarzi et al. in paper [19] proposed a RAS by categorizing the cluster in the system based on the number and type of computing, data storage and communication resources that they control. All of these resources are allocated within each server. The disk resource is allocated based on the constant need of the clients and other kind of resources in the servers and clusters are allocated using the Generalized Processor Sharing (GPS). This system performs distributed decision making to reduce the decision time by parallelizing the solution and used a greedy algorithm to find the best initial solution. The solution could be improved by changing resource allocation. But this system cannot handle large changes in the parameters which are used for finding the solution.

***Auction***

Cloud resource allocation by auction mechanism is addressed by Wei-Yu Lin et al. in [23]. The proposed mechanism is based on sealed-bid auction. The cloud service provider collects all the users' bids and determines the price. The resource is distributed to the first $k^{th}$ highest bidders under the price of the $(k+1)^{th}$ highest bid. This system simplifies the cloud service provider decision rule and the clear cut allocation rule by reducing the resource problem into ordering problem. But this mechanism does not ensure profit maximization due to its truth telling property under constraints.

The aim of the resource allocation strategy is to maximize the profits of both the customer agent and the resource agent in a large datacenter by balancing the demand and supply in the market. It is achieved by using market based resource allocation strategy in which equilibrium theory is introduced (RSA-M) [24]. RSA-M determines the number of fractions used by one VM and can be adjusted dynamically according to the varied resource requirement of the workloads. One type of resource is delegated to publish the resource's price by resource agent and the resource delegated by the customer agent participates in the market system to obtain the maximum benefit for the consumer. Market Economy Mechanism is responsible for balancing the resource supply and demand in the market system.

***Application***

Resource Allocation strategies are proposed based on the nature of the applications in [25]. In the work by Truong et al. [25], Virtual infrastructure allocation strategies are designed for workflow based applications where resources are allocated based on the workflow representation of the application. For work flow based applications, the application logic can be interpreted and exploited to produce an execution schedule estimate. This helps the user to estimate the exact amount of resources that will be consumed for each run of the application. Four strategies such as Naive, FIFO, Optimized and services group optimization are designed to allocate resources and schedule computing tasks.

Real time application which collects and analyzes real time data from external service or applications has a deadline for completing the task. This kind of application has a light weight web interface and resource intensive back end. To enable dynamic allocation of cloud resources for back-end mashups, a prototype system is implemented and evaluated for both static and adaptive allocation with a test bed cloud to allocate resources to the application. The system also accommodates new requests despite a-priori undefined resource utilization requirements. This prototype works by monitoring the CPU usage of each virtual machine and adaptively invoking additional virtual machines as required by the system.

David Irwin et al. [26] have suggested the integration of high bandwidth radar sensor networks with computational and storage resources in the cloud to design end-to-end data intensive cloud systems. Their work provides a platform that

supports a research on a broad range of heterogeneous resources and overcomes the challenges of coordinated provisioning between sensor networks, network providers and cloud computing providers. Inclusion of non-traditional resources like Steerable sensors and cameras and stitching mechanisms to bind the resources are the requirement of this project. Resource allocation strategy plays a significant role in this project.

The database replicas allocation strategy is designed in [27]. In that work, the resource allocation module divides the resource (CPU, Memory and DB replicas) allocation problem in two levels. The first level optimally splits the resources among the clients whereas the database replicas are expandable (dynamically) in the second level, based on the learned predictive model. It achieves optimal resource allocation in a dynamic and intelligent fashion.

***SLA***

In the cloud, the works related to the SaaS providers considering SLA are still in their infancy. Therefore, in order to achieve the SaaS providers' objective, various RAS specific to SaaS in cloud has been proposed. With the emergence of SaaS, applications have started moving away from PC based to web delivered-hosted services. Most of the RAS for SaaS focused towards customer benefits. Popovivi et al. [27] have mainly considered QoS parameters on the resource provider's side such as price and offered load.

Moreover Lee at.al [28] have addressed the problem of profit driven service request scheduling in cloud computing by considering the objectives of both parties such as service providers and consumers. But the author Linlin Wu et al. [29] have contributed to RAS by focusing on SLA driven user based QoS parameters to maximize the profit for SaaS providers. The mappings of customer requests into infrastructure level parameters and policies that minimize the cost by optimizing the resource allocation within a VM are also proposed in [29].

Managing the computing resources for SaaS processes is challenging for SaaS providers [30]. Therefore a framework for resource management for SaaS providers to efficiently control the service levels of their users is contributed by Richard et al. [30]. It can also scale SaaS provider application under various dynamic user arrivals/ departures. All the above mentioned mainly focus on SaaS providers' benefits and significantly reduce resource waste and SLO violations.

**Advantages and limitations**

There are many benefits in resource allocation while using cloud computing irrespective of the size of the organization and business markets. But there are some limitations as well, since it is an evolving technology. Let's have a comparative look at the advantages and limitations of resource allocation in the cloud.

**Advantages:**

1. The biggest benefit of resource allocation is that user neither has to install software, nor hardware to access the applications, to develop the application and to host the application over the internet.
2. The next major benefit is that there is no limitation of place and medium. We can reach our applications and data anywhere in the world, on any system.
3. The user does not need to expend on hardware and software systems.
4. Cloud providers can share their resources over the internet during resource scarcity.

**Limitations**

1. Since users rent resources from remote servers for their purpose, they don't have control over their resources.
2. Migration problem occurs, when the users wants to switch to some other provider for the better storage of their data. It's not easy to transfer huge data from one provider to the other.
3. In public cloud, the clients' data can be susceptible to hacking or phishing attacks. Since the servers on the cloud are interconnected, it is easy for malware to spread.
4. Peripheral devices like printers or scanners might not work with cloud. Many of them require software to be installed locally. Networked peripherals have lesser problems.
5. More and deeper knowledge is required for allocating and managing resources in the cloud, since all knowledge about the working of the cloud mainly depends upon the cloud service provider.

In the below table various resource allocation strategies and their impact are listed.

| S.No. | Resource Allocation Strategy | Impacts |
|---|---|---|
| 1 | Based on the estimated execution time of job. (Advanced Reservation, Best effort and immediate mode) | The estimation may not be accurate. If job could not finish its execution in estimated tine, it will affect the execution of other jobs. |
| 2 | Matchmaking strategy based on Any-Schedulability criteria. | Strategy mainly depends upon the user estimated job execution time of a job. |
| 3 | Based on the role based security policy. | Follows decentralized resource allocation. |
| 4 | Most Fit Processor Policy. | Requires complex searching process and practical to use in real |

| | | |
|---|---|---|
| | | system. |
| 5 | Based on the cost and speed of VM. | Allows the user to select VM. |
| 6 | Based on the load conditions specified by the user. | Instances of resources can be added or removed. |
| 7 | Based on the gossip protocol (resources allocated by getting information for other local nodes) | It used, decentralized algorithm to compute resource allocation and this prototype is not acceptable for heterogeneous cloud environment. |
| 8 | Utility function as a measure of profit based on live VM migration. | Focused on scaling CPU resources in IaaS. |
| 9 | Based on the utility function as a measure of price. | Allocate resources only at the lowest level of cloud computing and considered only CPU resource. |
| 10 | Utility function as a measure of response time. | Lacks in handling dynamic client requests. |
| 11 | Based on utility function as a measure of application satisfaction. | Relies on two-tier architecture. |
| 12 | Based on the CPU usage of VM, active user requests are served. Adaptively new VM spawns, when the CPU usage reaches some critical point. (VR) | There is a limitation in the number of concurrent user monitor and the prototype is not capable of scaling down as the number of active user decreases. |
| 13 | Based on hardware resource dependency. | Considered only CPU and I/O resource. |
| 14 | Auction mechanism. | Not ensure profit maximization |
| 15 | Based on online resource demand predication. | Prediction may not be accurate and leads to over provisioning or under provisioning. |
| 16 | Based on work flow representation of the application. | The application logic can be interpreted and exploited to produce an execution schedule estimate. Again estimation may not be accurate. |
| 17 | Based on the machine learning technique to precisely make decisions on resources. | This prototype reduces the total SLA cost and allocate resources considering the both the request rates and also the weights. |
| 18 | Simulated annealing algorithm. | Lacks in handling a dynamic resource request. |
| 19 | Based on constant needs of client and GPS. | The solution can be improved by changing the resource allocation and lacks in handling the large changes in parameters. |
| 20 | Stochastic approximation approach. | Very complex in nature. |
| 21 | Network game theory approach. | Lack in dynamic cooperative organization formation. |

## CONCLUSION

Cloud computing technology is increasingly being used in enterprises and business markets. In cloud paradigm, an effective resource allocation strategy is required for achieving user satisfaction and maximizing the profit for cloud service providers. This paper summarizes the classification of RAS and its impacts in cloud system. Some of the strategies discussed above mainly focus on CPU, memory resources, but are lacking in some factors. Hence this survey paper will hopefully motivate future researchers to come up with smarter and secured optimal resource allocation algorithms and framework to strengthen the cloud computing paradigm.

## REFERENCES

[1] Zhao, Ming, and Renato J. Figueiredo. "Experimental study of virtual machine migration in support of reservation of cluster resources." Proceedings of the 2nd international workshop on Virtualization technology in distributed computing. ACM,2007.

[2] Sotomayor, Borja, et al. "Resource leasing and the art of suspending virtual machines." High Performance Computing and Communications, 2009. HPCC'09. 11th IEEE International Conference on. IEEE, 2009.

[3] Sotomayor, Borja, Kate Keahey, and Ian Foster. "Combining batch execution and leasing using virtual machines."Proceedings of the 17th international symposium on High performance distributed computing. ACM, 2008.

[4] Li, Chunlin. "Optimal resource provisioning for cloud computing environment." The Journal of Supercomputing 62.2 (2012): 989-1022.

[5] Salehi, Mohsen Amini, Bahman Javadi, and Rajkumar Buyya. "Resource provisioning based on preempting virtual machinesin distributed systems." Concurrency and Computation: Practice and Experience 26.2 (2014): 412-433.

[6] Salehi, Mohsen Amini, BahmanJavadi, and Rajkumar Buyya. "Resource provisioning based on lease preemption in InterGrid." Proceedings of the Thirty-Fourth Australasian Computer Science Conference-Volume 113. Australian Computer Society, Inc., 2011.

[7] http:// Haizea.cs.uchicago.edu/

[8] Li, Jiayin, et al. "Adaptive resource allocation for preemptable jobs in cloud systems." 2010 10th International Conference on Intelligent Systems Design and Applications. IEEE, 2010.

[9] Melendez, Jose Orlando, and Shikharesh Majumdar. "Matchmaking with limited knowledge of resources on clouds and grids." Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2010 International Symposium on . IEEE, 2010.

[10] Shin, Dongwan, and Hakan Akkan. "Domain-based virtualized resource management in cloud computing." Collaborative Computing: Networking, Applications and Work sharing (CollaborateCom), 2010 6th International Conference on. IEEE,2010.

[11] Goudarzi, Hadi, and Massoud Pedram. "Maximizing profit in cloud computing system via resource allocation." 2011 31stInternational Conference on Distributed Computing Systems Workshops. IEEE, 2011.

[12] Ruth, Paul, et al. "Autonomic live adaptation of virtual computational environments in a multi-domain infrastructure." 2006 IEEE International Conference on Autonomic Computing. IEEE, 2006.

[13] Kumar, Karthik, et al. "Resource allocation for real-time tasks using cloud computing." Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on. IEEE, 2011.

[14] Kong, Zhen, Cheng-Zhong Xu, and Minyi Guo. "Mechanism design for stochastic virtual resource allocation in non- cooperative cloud systems." Cloud Computing (CLOUD), 2011 IEEE International Conference on. IEEE, 2011.

[15] Wuhib, Fetahi, and Rolf Stadler. "Distributed monitoring and resource management for large cloud environments." 12thIFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops. IEEE, 2011.

[16] Yang, Jie, JieQiu, and Ying Li. "A profile-based approach to just-in-time scalability for cloud applications." 2009 IEEE International conference on Cloud Computing. IEEE, 2009.

[17] Zhu, Xiaoyun, et al. "1000 islands: Integrated capacity and workload management for the next generation data center."Autonomic Computing, 2008. ICAC'08. International Conference on. IEEE, 2008.

[18] Minarolli, Dorian, and Bernd Freisleben. "Utility-based resource allocation for virtual machines in cloud computing."Computers and Communications (ISCC), 2011 IEEE Symposium on. IEEE, 2011.

[19] Van, Hien Nguyen, Frederic Dang Tran, and Jean-Marc Menaud. "SLA-aware virtual resource management for cloud infrastructures." Computer and Information Technology, 2009. CIT'09. Ninth IEEE International Conference on . Vol. 1. IEEE, 2009.

[20] Hu, Weisong, et al. "Multiple-job optimization in mapreduce for heterogeneous workloads." Semantics Knowledge and Grid(SKG), 2010 Sixth International Conference on. IEEE, 2010.

[21] Lu, Xiaoyi, et al. "Vega Ling Cloud: a resource single leasing point system to support heterogeneous application modes on shared infrastructure." 2011 IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications . IEEE, 2011.

[22] Endo, Patricia Takako, et al. "Resource allocation for distributed cloud: concepts and research challenges." IEEE network25.4 (2011): 42-46.

[23] Lin, Wei-Yu, Guan-Yu Lin, and Hung-Yu Wei. "Dynamic auction mechanism for cloud resource allocation." Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on. IEEE, 2010.

[24] You, Xindong, et al. "RAS-M: Resource allocation strategy based on market mechanism in cloud computing." 2009 FourthChinaGrid Annual Conference. IEEE, 2009.

[25] Truong Huu, T., and J. Montagnat. "Virtual resources allocation for workflow-based applications distribution on a cloud infrastructure." 2nd International Symposium on Cloud Computing (Cloud 2010), Melbourne, Australia. IEEE Computer Society. 2010.

[26] Irwin, David, et al. "Resource management in data-intensive clouds: opportunities and challenges." Local and MetropolitanArea Networks (LANMAN), 2010 17th IEEE Workshop on. IEEE, 2010.

[27] Popovici, Florentina I., and John Wilkes. "Profitable services in an uncertain world." Proceedings of the 2005 ACM/IEEEconference on Supercomputing. IEEE Computer Society, 2005.

[28] Lee, Young Choon, et al. "Profit-driven service request scheduling in clouds." Proceedings of the 2010 10th IEEE/ACMinternational conference on cluster, cloud and grid computing. IEEE Computer Society, 2010.

[29] Wu, Linlin, Saurabh kumar Garg, and Rajkumar Buyya. "SLA based resource allocation for software as a service provider (saas) in cloud computing environments." Cluster, cloud and grid computing (CCGrid), 2011, 11th IEEE/ACm international symposium on. IEEE, 2011.

[30] Richard T.B.Ma,Dah Ming Chiu and John C.S.Lui, Vishal Misra and Dan Rubenstein:On Resource Management for Cloud users :a Generalized Kelly Mechanism Approach.