STUDY ON SCALABILITY SERVICES IN CLOUD COMPUTING

C.Venish Raja[1], Dr.L. Jayasimman[2],
[1]Research Scholar, Department of Computer Science, Bishop Heber College, Trichy, Tamilnadu, India.
[2]Assistant Professor, Department of Computer Science, Bishop Heber College, Trichy, Tamilnadu, India.

Abstract: Cloud computing refers an automated on-demand self-service, that allows a pay-per-use model on shared resources. Different field especially for business in companies are relying on cloud storage. Scalability is considered a significant factor for securing the cloud environment to overcome the hacking problem. In cloud paradigm, scalability is that one of the major benefits and, particularly, which makes different to an "advanced outsourcing" solution. Among all the exciting feature of cloud computing is Auto-Scaling that recommends the clients the comfort and easy to utilize the resources as per their demand and expectations. This paper emphasizes the auto scaling method and presents an outline of cloud computing. In this paper, a study of cloud scalability, issues of scalability, auto scaling techniques and their algorithms such as a proactive auto-scaling algorithm (PASA) and a dynamic auto-scaling algorithm (DASA).

Keywords: Cloud scalability, Scalability issues, Scalability Factor, Scalability Services, Horizontal Scalability, Vertical Scalability, Auto Scaling, Auto Scaling techniques, Auto Scaling Algorithms

I. INTRODUCTION

Cloud Computing is effectively powerful computing paradigm to deliver services over the internet. It is a model to facilitate or enable on-demand network access, on-demand self-service convenient to a shared pool of computing resources in configurable manner which can be quickly provisioned. The model of Cloud Computing has been differentiated into IaaS (infrastructure-as-a-service), SaaS (software-as-a-service) and Paas (Platform-as-a-service) [1]. Cloud Storage is a service, in which the data is remotely managed, backed up, maintained and restore and it makes data available to users via internet or network. Several cloud storage providers provide free space up to certain gigabytes. For example, Drop Box offer free space up to 2GB, Google Drive, Amazon, Apple Cloud make available free space up to 5GB, Microsoft Sky-Drive give free space up to 10 GB [2].

One of the key advantages of using cloud computing paradigm is called as scalability. It supports the long term strategies and business needs and is entirely different than elasticity. It is the mechanism by which clients dynamically provision their resources such as software applications and hardware devices even if demand and situation arise like that. Cloud provides elasticity by scaling up as computing must increase after that scaling down again as demands decrease. Auto scaling is that enables users to automatically scale up or down based on-demand self-service in cloud computing. In Cloud Computing environment, the virtualization technology [3, 4, and 5] plays the important role to provision the physical resources, like disk storage, processors, and broadband network. Virtualization refers primarily to platform virtualization for users. A VNF (Virtualized Network Function) instance running on VM (Virtual Machine) can scale-out/in (turn on/off) to adjust the VNF's computing and networking capabilities, consuming on both resources and energy. Tools that automatically change the amount of used resources are known as "auto-scaling services".

Auto-scaling is the technique that has ability to adjust the available resources to meet the user expectations and demands. In this paper, the cost-performance tradeoff while considering both the legacy equipment capacity and VM setup time. A category of auto-scaling techniques are classified into five aspects, namely threshold-based rules (rules), time series analysis, queuing theory, control theory and reinforcement learning. To solve the problem using Dynamic Auto Scaling Algorithm (DASA) and Proactive Auto Scaling Algorithm (PASA) predict the workload ahead such that the auto-scalar can make decision based on the expected workload instead of waiting for trigger. In this paper, an analysis of auto-scaling concepts and techniques, and research challenges. The ultimate goal is to recognize the auto-scaling services and future research works.

II. CLOUD SCALABILITY

In the area of scalability is concentrated on scaling strategies and algorithms aiming at maximizing the performance metrics and minimizing the related costs or on architectures that must be applied to ensure that the application would effectively scale [6]. Another definition state that [7] "Scalability is the ability of an application to be scaled up to meet demand through replication and distribution of requests across a pool or farm of servers".

"Scalability concept is ability of a system to accommodate an increasing number of elements or objects, to process growing volumes of work gracefully, and/or to be susceptible to enlargement" [8]. Four kinds of measurements are used in scalability as follows:

- **Load scalability (LS):** It is capable of operating graceful different loads while making better usage of available resources. A few factors that can hamper load scalability is scheduling of a class of resources and scheduling of a shared resource in a way that enhances its inadequate exploitation of parallelism and own usage.
- **Space scalability:** It refers to the enlargement of memory usage compared to the scale of the system. Many different approaches like space efficient algorithms and compression can help with space scalability, but the effects (like added CPU time of compression) might reduce other types of scalability like load scalability.
- **Space-time scalability:** It regards the ability of a system functions gracefully improve when the number of items it handles increase by an order of magnitude. Space-time scalability may be related to both space

scalability and load scalability in that the amount of items might stem from an increased load, and the presence of these objects may use more memory and affect data structures.

- **Structural scalability:** It refers to the standards of the system and how they limit the number of item the system may handle. The prime example of structural scalability concerns the addressing of the items, for instance will a fixed addressing space put a limit on the systems scalability.

Scalability issues in Cloud

In cloud environment, scalability is the promising task for effective usage of the resources and to improve the profit of CSP (Cloud Service Providers). In recent years, the companies and enterprises are tried to attain the scalability in terms of platform, application, and database and infrastructure level. The ability of a particular system performs to fit as the scope of that issue increases (number of objects or elements, rising volumes of work and/or being susceptible to enlargement) [9]. Also can defined as Scalability of service is a desirable property of a service which provides an ability to handle growing amounts of service loads without suffering significant degradation in relevant quality attributes [10].

In cloud computing, multi-tenancy is the concept in each level to solve the scalability issues in an efficient manner. In SaaS multi-tenancy is defined as "a single application instance shared by multiple customers. In PASA (Proactive Auto Scaling Algorithm) multi-tenancy is defined as "a single platform/container instance capable of handling or deploying different type of applications. In DASA (Dynamic Auto Scaling Algorithm) multi-tenancy is defined as "a single database and single schema that is shared by multiple organizations/tenants".

Scalability Factor

During scaling time, it is significant to note that what percentage of resource is actually scalable. It is called as **scalability factor**. Each component whether it is servers, processors, load-balancers or storage, which is to be scaled some kind of management overhead. For example, when lose 5% of the processor power every time we add a CPU to the system, then the scalability factor is 0.95.

Four categories of scalability factor are described as follows:

- **Linear scalability:** It performs scalability that remains constant inspite of scaling.
- **Sub-linear scalability:** At this point scalability factor reduces below 1.0.
- **Supra-linear scalability:** It is possible to obtain better throughput performance by adding one resource (which is very rare case) is known as supra-linear scalability.
- **Negative scalability:** If the performance of an application degrades when the application is scaled which is known as negative scalability.

SCALABILITY SERVICES

In cloud paradigm, scalability is the major advantages to be effectively performed. Particularly, it provides clouds from advanced outsourcing solutions. Other than, a few significant pending problems should be addressed before the dream of automated scaling of applications can be realized. The most notable initiatives towards whole application scalability in cloud environments are given below [11].

Server Scalability

Most of the accessible IaaS clouds handle single VM management primitives (e.g., elements for adding/ eliminating VMs), lacking mechanisms for treating applications as an entire single entity and coping with the relations between different application components. Therefore, the database requires initially to be deployed, it obtains IP and configures the Web server connecting to it. In general, application providers are needed to handle their application only, being discharged from the burden of dealing with (virtual) infrastructure terms.

Scaling the Network

In general, networking over virtualized resources is considered into two different manners such as "Ethernet virtualization" and TCP/IP virtualization. Using these techniques are mainly focused in the usage of VLAN (virtual local area network) tags L2 (Data Link layer) to separate traffic or public key infrastructures to make L2/L3 (Network Layer) overlays [12, 13, 14, 15]. This method is expensive and induces network instability at the same time as the infrastructure is being updated. Moreover, it is static type and does not take into account that not all the applications consume all the required bandwidth during that time. Improved techniques taking into account actual network usage are necessary.

Scaling the Platform

IaaS clouds are useful for application providers to manage the resources utilized by their systems. But, IaaS clouds demand application developers or system administrators to install and configure all the software stacks the application components require. To compare that PaaS clouds provide a ready to make use of execution environment, along with convenient services, for applications. Therefore, when using PaaS clouds developers can focus on programming their components rather than on setting up the environment those components need. It can be only possible to handle scalability issues of all the services in a superficial manner. Instead, a most analysis of the container, the important services, and the database, has been preferred.

III. CLOUD SCALABILITY TYPES

Horizontal Scalability

Horizontal cloud scalability is mainly used the ability to connect multiple hardware or software entities, like servers, hard drives in order that they work as a single logical unit. Horizontal scalability is provided by means of adding or removing more individual units of resource doing the same job. In the case of servers, you could increase the speed or availability of the logical unit by adding more servers. This is the most common way of scaling and also the cheapest. Horizontal scaling (out) requires the addition of more devices or machines to the computing platform to handle the increased demand. It is also referred to as scaling out. On the other hand, horizontal scaling is needed to restore and maintain peak performance. It is also time consuming and manually intensive, requiring a technician to add machinery to the customer's cloud configuration.

Vertical Scaling

Vertical scalability is the ability to maximize the capacity of existing hardware or software by adding more resources to the hardware or same server. Improving the vertical scalability is important in achieving the low investment on cloud computing and virtualization. Scaling up involves adding more resources to the same computing pool, (e.g., adding more RAM, disk, or virtual CPU to handle an increased application load).

In vertical scaling replace the current IT resource by another one with higher capacity (scaling up) or with lower capacity (scaling down). Hence, it enhances the capacity of existing hardware and software. It is the ability of the application to be scaled under load.

For instance, adding processing power to a server to make it more rapidly [7]. This kind of scaling is more expensive and less common. It can be achieved through the addition of extra hardware to the same entity such as hard drives, servers, CPU's, etc. It offers more shared resources for applications and operating system. This kind of scalability process can also be configured to as scaling in. It is also slower than horizontal scaling because of the downtime required during the replacement of the resource.

IV. AUTO SCALING

In cloud computing, Auto-Scaling (AS) is that allow user to automatic scale cloud services, such as Virtual Machine (VM) and server capacities Up or Down, depending on defining situation. Auto-scaling automates the contraction of system capacity that is available for applications and is a desired feature in cloud PaaS and IaaS offerings. When feasible, technology buyers should use it to match provisioned capacity to application demand and reduce costs. In Amazon Web Service (AWS), auto-scaling is defined as a cloud computing service feature that allows AWS users to automatically launch or terminate virtual instances based on defined policies, health status checks, and schedules. Auto-scaling is the capability in cloud computing infrastructures that enables dynamic provisioning of virtualized resources. Resources used by cloud based applications can be automatically maximized or minimized, in that way adapting resource usage to the application requirements [16].

Auto Scaling ensures that the correct number of EC2 instances available to handle or perform the load for application. If they generates collections of EC2 instances, known as Auto Scaling groups. In each Auto Scaling group, both the minimum number of instances and the maximum number of instances is separately specified, and AS ensures that group never goes below and above sizes. If specify the desired capacity, either when the group is generated at any time thereafter, AS ensures that the group has several instances. Even if it specifies scaling policies, after that AS can terminate or launch instances as demand on that application increases or decreases [16]. Auto scaling group has illustrated a maximum size of 4 instances, a minimum size of 1 instance, and a desired capacity of 2 instances as given below. The scaling policies define adjust the number of instances that handles maximum and minimum number of instances [16].

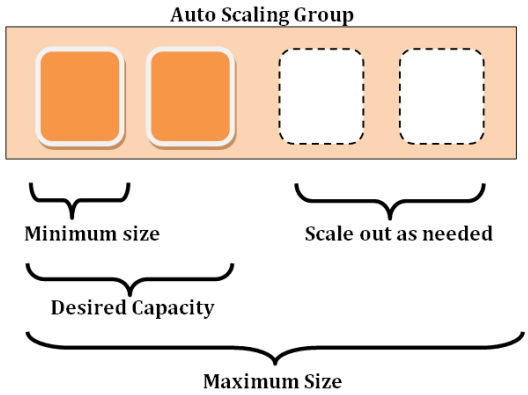

Figure 1: Auto Scaling

The key features of AS are based on these definitions as follows:

- The ability to scale out (i.e., the automatic addition of extra resources during demand increases) and scale in (i.e., the auto-terminate of extra unused resources when demand decreases, so as to minimize cost).
- AS can dynamically raise and reduce capacity as required and it pays for the EC2 instances to use and minimize cost by launching instances when they are actually needed and terminating them if they aren't required.
- To automatically detect and replace unreachable instances.
- Better fault tolerance: AS can detect when an instance is unreachable, terminate it, and launch an instance to replace it.
- Better availability: AS is to utilize and configure multiple Availability Zones, if it becomes unavailable, AS can launch instances in another one to compensate.

AUTO SCALING TECHNIQUES

1. Threshold-based rules (rules)
2. Reinforcement learning (RL)
3. Queuing theory (QT)
4. Control theory (CT)
5. Time series analysis (TS)

Threshold-based rules:

Commercial cloud providers give purely reactive AS using threshold-based rules. The scaling decisions are triggered based on some performance metrics and predefined thresholds. This approach has become rather popular due to its (apparent) simplicity: rule-based auto-scaler is also easy to set-up by clients, and are easy to provide as a cloud service. But, the effectiveness of rules under bursty workloads is questionable.

Time series analysis (TSA)

TSA covers an extensive range of techniques to detect patterns and predict future values on sequences of data points. The accuracy in the forecast value (e.g. future number of requests or average CPU utilization) will depend on chosen the right technique and setting the parameters. An example is the number of requests that reaches an application, taken at one-minute intervals. The time-series analysis could be used to find repeating patterns in the input workload or to try to forecast future values. Time-series analysis is the main enabler of proactive auto-scaling techniques. There are two auto-scaling methods that rely on modeling the system in order to

determine its future resource needs. This is the case of both queuing theory and control theory.

Queuing theory (QT)
Queuing theory has been mainly applied to computing systems, in order to discover the relationship between the jobs arriving and leaving a system. Queuing theory can be used to add capacity by analyzing and making decisions based on a queue specifically requests queued at the load balancer. A simple approach consists in modeling each VM as a queue of requests in order to estimate different performance metrics such as the response time. Since queuing theory only provides an estimation of performance metrics, have combined with another approaches (i.e., threshold based policies, control theory, and reinforcement learning) to deal with auto-scaling problem. There are two important obstacles to using queuing theory approaches in auto-scaling systems. First, they impose non-realistic assumptions that are not valid in real scenarios; and second, they are not efficient for complex systems.

Control theory (CT)
Control systems utilize a feedback loop by changing the controller input to influence the normative output. The aim is to define a (reactive or proactive) controller to automatically adjust the required resources to the application demands. Control systems are mainly used as reactive process, but there are also some proactive approximations like Model Predictive Control, or even combining a control system with a predictive model. CT has been applied to automate management of resources in different engineering fields, like data centers, storage systems, and cloud computing platforms. The objective of a controller is needed to maintain the output of the target system (e.g., performance of a cloud environment) to a desired level by adjusting the control input (ex: number of VMs). This kind of auto-scaling has a great potential, especially when combined with resource prediction.

Reinforcement learning (RL)
Auto-scaling based on reinforcement learning is a predictive method to auto-scaling. At last technique, the last of our categories contains proposals based on reinforcement learning. Similarly to CT, RL tries to automate the scaling task, but without using any Apriority knowledge or model of the application. RL tries to learn the most suitable action for each particular state on-the-fly, with a trial-and-error approach. The main disadvantages of these approaches are bad initial performance, and the problem to handle and perform sudden bursts in input workload. The time needed by the method to converge to an optimal policy can be unfeasible long. In the cloud provisioning problem domain, the auto-scaling component is the agent that interacts with the scalable application environment and decides whether to add or eliminate resources to increase the maximum award (i.e., minimize response time).

V. AUTO SCALING ALGORITHMS

The goal of VNF instance auto-scaling algorithm is to reduce operation cost while providing acceptable levels of performance. Auto-scaling techniques are diverse, and accommodate different components at the platform, infrastructure, and software services. The main aspects of PASA and DASA are clearly described as follows.

PASA (PROACTIVE AUTO SCALING ALGORITHM)
The main challenge of proactive auto scaling is that workload prediction. It can be done by numerous methods such as Threshold based, Reinforcement learning (RL), Queuing theory, Control theory, Time series Analysis (TSA), which are already discussed in above section. Using TSA based technique predicts future values and detects pattern of workload. There are several methods depend upon TSA for prediction such as Autoregressive (AR), Moving Average (MA), Autoregressive moving average (ARMA), Autoregressive integrated moving average (ARIMA) etc.

DASA
Dynamic Auto Scaling Algorithm (DASA) is referred the tradeoff between operation cost and performance maintenance. To improve an analytical model is effectively to validate and quantify the tradeoff analysis through extensive simulations. The performance is evaluated by average response time per user request. More power-on VNF instances minimize the possibility of SLAs violations. But, this may incur redundant power-on VNF instances, leading to more operation cost. Initially, propose DASA algorithm to balance the tradeoff issues.

When failures like this occur, an auto-scaling mechanism needs to recover in an intelligent way using PASA and DASA.

VI. CONCLUSION

Developments in the area of auto-scaling platforms are specifically important for micro-service applications however traditional cloud-based systems would also benefit from existence of open-source auto-scaling platforms. The main objective of this paper is to present a comprehensive study about the auto-scaling mechanisms available today, in addition to highlight the open issues in the field. In this paper, a careful analysis of current state of auto-scaling in cloud computing. Here, various types of scaling, services of stability, some factors, and algorithms were discussed briefly. Developing the performance of the cloud systems are increased by the auto-scaling technique; this is due to the detail that, some mechanisms have been proposed for auto scaling.

VII. REFERENCE

[1]. R. Prodan and S. Ostermann, Proceedings of the 10th IEEE/ACM international conference on Grid Computing, (2009) October 13-15; Banff, Canada.
[2]. R.Aronika Paul Rajan, S. Shanmugapriya (2012,May-Jun).Evolution of Cloud Storage as Cloud Computing Infrastructure Service. IOSRJCE.1 (1),pp-38-45.
[3]. VMwareInc., http://www.vmware.com/products/vi/esx/.
[4]. Xen, http//www.xen.org.
[5]. Kernel-based Virtual Machine (KVM), http://www.linux-kvm.org.
[6]. M.A.S. Netto, C. Cardonha, R.L.F. Cunha and M.D. Assuncao, "Evaluating auto-scaling strategies for cloud computing environments", in Proceedings - IEEE Computer Society's Annual International Symposium 2014,pp.187,196. https://doi.org/10.1109/MASCOTS.2014.32 on
[7]. T.Chieu, A.Mohindra and A. Karve, "Scalability and Performance of Web Applications in a Compute Cloud," in e-Business Engineering (ICEBE), 2011.

[8]. André B. Bondi, Characteristics of scalability and their impact on performance, In Proceedings of the 2Nd International Workshop on Software and Performance, WOSP '00, pages 195–203, New York, NY, USA, 2000. ACM. ISBN 1-58113- 195-X. doi: 10.1145/350391.350432.
[9]. B.Furht and A.Escalante, in Hand Book of Cloud Computing, Springer, 2010.
[10]. J.Lee and S. Kim,"Software Approaches to Assuring High Scalability in Cloud Computing,"in IEEE Inter national Conference on E-Business Engineering, 2010.
[11]. L.Vaquero, L.Rodero-Merino and R. Buyya, "Dynamically Scaling Applications in the Cloud," ACM SIGCOMM Computer Communication Review, pp. 45-52, January 2011.
[12]. X. Jiang and D. Xu, "Violin: Virtual internetworking on overlay infrastructure," in *PDPA03: Proceedings of the 2nd International Symposium on Parallel and Distributed Processing and Applications*, 2003, pp. 937–946.
[13]. A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford, "In vini veritas: realistic and controlled network experimentation," in *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2006, pp. 3–14.
[14]. I. Baldine, Y. Xin, D. Evans, C. Heerman, J. Chase, V. Marupadi, and A. Yumerefendi, "The missing link: Putting the network in networked cloud computing." in *ICVCI09: International Conference on the Virtual Computing Initiative*, 2009.
[15]. T. W. Alex, P. Shenoy, and J. V. Merwe, "The case for enterprise-ready virtual private clouds," in *HotCloud09: Proceedings of the Workshop on Hot Topics in Cloud Computing*., 2009, pp. 1–5.
[16]. "Amazon Auto Scaling in Cloud Computing", http://aws.amazon.com/autoscaling/30.05.2012

Authors Profile

Author I

Mr.C.Venish raja working as a Assistant Professor in the Department of Information Technology, St.Joseph's College (Autonomous) Trichy, India.

He received his M.Phil Degree in Bharathidasan University, Trichy, India in 2012 and also He is pursuing Ph.D (Computer Science) in Bharathidasan University.

Author II

Dr. L. Jayasimman working as a Assistant Professor in the Department of Computer Science, Bishop Heber College, Trichy, India.

He received his M.Tech Degree in Bharathidasan University, Trichy, India in 2008 and completed his PhD (Computer Science) Bharathidasan University in 2014.