A. TAYEB

ATILIM UNIVERSITY 2019

A LITERATURE SURVEY ABOUT VERTICAL AND HORIZONTAL SCALABILITY IN CLOUD COMPUTING

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

ABDULSALAM SALIM TAYEB TAYEB

A MASTER OF SCIENCE THESIS
IN
THE DEPARTMENT OF INFORMATION SYSTEM ENGINEERING

JUNE 2019

A LITERATURE SURVEY ABOUT VERTICAL AND HORIZONTAL SCALABILITY IN CLOUD COMPUTING

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

OF

ATILIM UNIVERSITY

BY

ABDULSALAM SALIM TAYEB TAYEB

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SIENCES

IN

INFORMATION TECHNOLOGY

JUNE 2019

Approval of the Graduate School of Natural and Applied Sciences, Atilim University.

____

Prof. Dr. Ali KARA
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Master of Science in Information Technology, Atilim University**.

____

Assoc. Prof. Dr. Korhan Levent ERTÜRK
Head of Department

This is to certify that we have read the thesis A LITERATURE SURVEY ABOUT VERTICAL AND HORIZONTAL SCALABILITY IN CLOUD COMPUTING submitted by ABDULSALAM SALIM TAYEB TAYEB and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

____

Assoc. Prof. Dr. Murat KOYUNCU
Supervisor

Examining Committee Members:

Assoc. Prof. Dr. Nergiz ERÇİL ÇAĞILTAY
Software Eng. Department, Atilim University ____________________________

Assoc. Prof. Dr. Murat KOYUNCU
Information Eng. Department, Atilim University ____________________________

Asst. Prof. Dr. Erol ÖZÇELİK
Psychology Department, Çankaya University ____________________________

Date: 28 .06.2019

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

ABDULSALAM SALIM TAYEB TAYEB

ABSTRACT

A LITERATURE SURVEY ABOUT VERTICAL AND HORIZONTAL SCALABILITY IN CLOUD COMPUTING

TAYEB, Abdulsalam Salim Tayeb
MS., Information Technology
Supervisor: Assoc. Prof. Dr. Murat KOYUNCU

June 2019, 75 pages

Cloud computing has gained popularity in the industry due to the expansion of data generation from various domains such as physics, sciences, business and so on. Scalability can play an essential role in clouds and can improve their performance. The importance of scalability lies in its ability to cope with the increasing workloads by adding additional resources when demand increases. Typically, there are two types of scalability, namely horizontal scalability and vertical scalability. Hence, there are different classifications of scalability in the literature. This thesis is intended to be a comprehensive survey covering all available scalability techniques in cloud including different classifications based on different criteria. This study basically investigated vertical and horizontal scalability methods. In addition, it explores the used technologies for both of them. Moreover, it determines the amount of research publications related to scalability from different perspectives. To reach the specified goal, a systematic mapping study has been done about scalability.

Keywords: cloud computing, scalability, horizontal scalability, vertical scalability, systematic mapping.

ÖZ

BULUT BİLİŞİMDE DİKEY VE YATAY ÖLÇEKLENEBİLİRLİK KONUSUNDA BİR LİTERATÜR TARAMASI

TAYEB, Abdulsalam Salim Tayeb
YL., Bilişim Teknolojileri
Tez Yöneticisi: Doç. Dr. Murat KOYUNCU

Haziran 2019, 75 sayfa

Bulut bilişim, fizik, bilim, işletme ve benzeri çeşitli alanlarda veri üretiminin genişlemesi nedeniyle sektörde popülerlik kazanmıştır. Ölçeklenebilirlik bulutlarda önemli bir rol oynayabilir ve performanslarını artırabilir. Ölçeklenebilirliğin önemi, talep arttığında ek kaynaklar ekleyerek artan iş yükleriyle başa çıkma kabiliyetinde yatmaktadır. Genellikle, iki tür ölçeklenebilirlik vardır: yatay ölçeklenebilirlik ve dikey ölçeklenebilirlik. Dolayısıyla literatürde farklı ölçeklenebilirlik sınıflamaları vardır. Bu tez, farklı ölçütlere dayanan farklı sınıflandırmalar dahil olmak üzere buluttaki mevcut tüm ölçeklenebilirlik tekniklerini inceleyen kapsamlı bir araştırma olarak tasarlanmıştır. Bu çalışmada temel olarak dikey ve yatay ölçeklenebilirlik yöntemleri incelenmiştir. Ek olarak, her ikisi için de kullanılan teknolojiler araştırılmıştır. Ayrıca, ölçeklenebilirlik ile ilgili araştırma yayınları farklı açılardan değerlendirilmiştir. Belirlenen hedefe ulaşmak için ölçeklenebilirlik konusunda sistematik bir haritalama çalışması yapılmıştır.

Anahtar Kelimeler: bulut bilişim, ölçeklenebilirlik, yatay ölçeklenebilirlik, dikey ölçeklenebilirlik, sistematik haritalama.

To my family

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my supervisor Assoc. Prof. Dr. Murat KOYUNCU for his support, guidance, encouragement, useful critiques for this research work and patience during my research. I owe great thanks to him.

I would like to thank my parents, my wife and my brothers and sisters for their continuous support and patience during this period, none of this would have been possible without their support.

Furthermore, I thank my friend SAIF ALJUMAILI for his effort and helps.

Finally, I would like to thank my cousin ABDULVAHAP METE and his family for their support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Cloud computing has gained plenty of attentions in industrial and academic societies due to the expansion of data generation from various sectors such as physics, sciences, business and medicine. Accessing to big data becomes crucial for various applications such as combat of crimes, prevention of disease, urban traffic regulation, etc. Therefore, data faces real challenges related to storage, data acquisition and analysis. Parallel data computing is becoming more and more popular to tackle the challenges faced by big data. The same idea is enlightened by big technology companies such as Dryad and Google which provided a paradigm to scale big data among large clusters with eases in access facilities. In order to accommodate big and various types of data, the so-called “Cloud” is introduced.

The term Cloud is a type of technology, in which many systems are connected to each other in private or public networks, providing a scalable infrastructure for data, storage and application. Moreover, the cloud gives clients the ability to manage and control their application services, hence user can independently manage their own cloud applications and services. Development of clouds enables small organizations to access a big setup (services) with lesser cost. In another word, the cloud plays a vital role in resources utilization by the efficient distribution of services among a large number of customers. In order to understand cloud computing well, it is necessary to look at deployment models and static abstraction services.

Deployment models basically have four types, namely private, public, community and hybrid cloud. First, a private cloud is available only to organizations and exclusively to employees at the organizational level. Also, the organization controls and manages it. The public Cloud is available to all users in general and the organization that sells the Cloud services are responsible for their management and control. The community Cloud is shared by several organizations that share common interests, namely "mission, security requirements, policy considerations". The last model is hybrid Cloud in which two or more cloud types are mixed [1].

Cloud can provide different types of services, namely "Infrastructure-as-a-Service (IaaS) [2, 3], Software-as-a-Service (SaaS) [4], Platform-as-a-Service (PaaS) [5, 6]". In the SaaS model, the Cloud providers provide a software application that any company or user can purchase and use it on demand. Gmail can be an example of this type. The next service type is the PaaS model which has different types of environment including databases, operating systems, programming languages, web servers. In PaaS, consumers can create, run and deploy their applications. However, IaaS can provide machine, storage and network services to customer [7].

There are many benefits that can be obtained by using the Cloud over the Internet. The most prominent of these advantages is scalability which can be defined as the ability of the system to solve the problem of increasing the number of elements and increasing the service load and provides the ability to deal with these problems without the deterioration of quality features and to meet all requests through the replication of applications and distribution across a pool of servers [8].

Scalability can play an essential role in Cloud and can improve the performance of Cloud. Its importance lies in its ability to handle increasing workloads or its ability to improve work when resources are added. Also, how much the added resource is higher, the improvement is better. There are typically two types of scaling: horizontal scalability and vertical scalability [9].

Horizontal Cloud Scalability means that existing devices or resources have the ability to increase by adding other resources (connect multiple software or hardware entities), such as networks or servers that function as a single logical unit. This means adding resources that have the same job. For example, in case of servers, the speed of the logical unit could be increased by adding more servers as per the need. One can have one, two, nine or more servers which do the same work [10, 11].

Vertical scalability means the ability to increase the efficiency of existing programs and devices by adding additional new resources to the same hardware or server. For example, changing the CPU to a more efficient one than its predecessor in the server, adding hard drives and increasing the main memory capacity can be given as examples for vertical scalability. This provides additional common resources for the operating system and applications [10, 11].

Scalability is the utmost part of the Cloud since it can affect the cloud performance and computation time of the clients. Thus, the purpose of this thesis is to investigate the studies conducted related to scalability. As mentioned above, scalability can be achieved horizontally or vertically. Therefore, we aim to find out all the techniques proposed for both horizontal and vertical scalability types. The study is conducted as a systematic literature survey as explained in [61]. A systematic literature review is entirely based on secondary studies, such as scholarly articles, research papers, and published thesis. The main reason for this study is that there is a large amount of information about scalability in cloud which needs to be collected, analyzed, and eventually interpreted in a way as to provide compelling results for the present work. Therefore, the thesis is intended to be a comprehensive survey of all available scalability in cloud including different classifications based on their different criteria.

The introduction chapter of this thesis establishes the rationale for conducting this research. Chapter 2 gives general information about the cloud computing. Chapter 3 presents a literature survey about systematic mapping and cloud scalability. Chapter 4 gives the systematic mapping of the thesis, and finally, Chapter 5 provides conclusions.

CHAPTER 2

CLOUD COMPUTING

2.1 Definition

Cloud computing is a type of technology in which many systems are connected to each other as private or public networks, providing a scalable infrastructure for data, storage and application. After this technology has been appeared, through this technology, it becomes easy to access the information which is stored in the Cloud by any computer or any mobile device that can connect to the Internet at anytime and anywhere in the world.

Cloud computing has no clear definition, but the helpful one is the following definition by Forrester Research because of containing all the elements that Cloud computing have: “A standardized IT capability (services, software or infrastructure) delivered via Internet technologies in a pay-per-use, self-service way" [8]. According to National Institute of Standards and Technology (NIST) [9], Cloud Computing is: “a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”. Moreover, according to Amazon [10], Cloud computing has been defined as an on-demand providing of computing resources through the online Cloud platform and pay-per-use. Microsoft Azure [11] defines Cloud computing as “Cloud computing is the delivery of computing services servers, storage, databases, networking, software, analytics, intelligence and more over the Internet (“the Cloud”) to offer faster innovation, flexible resources, and economies of scale. You typically pay only for Cloud services you use, helping lower your operating costs, run your infrastructure more efficiently, and scale as your business needs change”.

2.2 Characteristics of Cloud Computing

There are many characteristics of Cloud computing, but there are five basic characteristics agreed by researchers. These are: [12, 13]

On-demand self-service: With on-demand self-service, consumer can control the available services in the Cloud, such as adding or cancelling services without needing to refer to the service providers, usually by interacting “Dashboard”.

Resource pooling: Different resources are collected in one physical location or in different physical locations according to conditions such as (security and consumer demand) to serve consumers depending on user demand. These resources can comprise storage, memory, processing, and network bandwidth. For legal reasons, users need the information about the location of datacenters but generally, the Cloud service users have no information about how and where service provider’s datacenters could maintain these resources [14].

Elasticity: This feature is one of the most important characteristics of Cloud computing. It gives Cloud computing a special advantage, it can quickly and flexibly provide resources, and in some cases automatically, also the ability to increase or decrease Cloud services on demand, without resorting to the service provider. In order to take advantage of the elasticity of a Cloud infrastructure, the applications should have the ability to scale up (adding more resources) and scale down (removing resources that are no longer required).[15]

Broad Network Access: The consumer can access through standard mechanisms and devices such as mobile phones, computers and operating systems. So, it is a flexible service - where laptops, computers and tablets can be available to the user as much or as little as he wants at any time and through any type of communication device.

Measured service: Providers and consumers have the ability to control, measure and report Cloud computing resources by using metering capability. The main purpose of the measured services is to economize based on the measurement as well as to pay according to what is consumed from the resources. This pricing model provides flexibility in the computation of resources and helps to provide efficient use and flexibility in traffic [16].

2.3 Advantages of Cloud computing

In this section we mention the advantages of Cloud computing.

Reduced costs: Instead of buying multiple servers and their own components, licensing operating systems and applications, installing all these and following up on updates and troubleshooting, the Cloud works centrally, and updates its components centrally without disrupting any Cloud. On the other hand, the calculation of the cost of using the Cloud depends on the amount of resource consumption, such as how much data processing, memory capacity, storage and data transfer is calculated by hour or month. In general, the larger the company and the larger the number of employees, the greater the resulting savings.

Larger storage capacity: You rarely find a server using even its maximum capacity, either in terms of processing or storage, so when you combine these unused energies together, on a large scale, you find a great abundance that can be exploited, the most prominent of which are giant storage capacities that can be accessed from more than one source. Studies and statistics indicate that businesses need to double their computing resources every year. At the top of the list comes the need for storage, as the volume of customer databases and users of any company increases.

Flexibility: With one command, a replica of an existing server running can be made or restored and run when the original server stops for any reason. In the same way, server resources can be increased with a sudden increase in the number of users, and server resources used by a limited number of users can be reduced to reduce their cost. Companies that need to experiment with various hardware and applications to find out which solution will suit their needs can do so on the Cloud in less time and at less cost than buying servers and testing applications.

Easy access: Whether you're browsing the Internet from a mobile, tablet or a small home computer, or a TV that supports Internet browsing, you can access data stored in the Cloud whether you're at home, on a plane, on a ship, or on a car. The Cloud makes you closer to your data and information, thus helping you to increase productivity and profitability.

Easier in management: Instead of wasting time managing each component of a company's server matrix, distributed over multiple locations, cities and countries, the Cloud provides a centralized control interface that controls each component of the Cloud, without having to travel or connect a screen to a server or otherwise. This interface can be accessed over the Internet. This centralized management makes servers much easier, less time and less tired, and one specialist can implement them, instead of providing a team of engineers and technicians going on each server and watching. Activating application updates and operating systems on the Cloud is easier and easier, because of this centralization.

More time: Because of all of the above, system administrators will have more time at their disposal, enabling them to think about future strategic planning for the company and the organization. This may have helped to reduce the technical staff, which in turn reduces costs and expenses and thus increases profitability and productivity.

Greater security: Because data is centrally stored in the Cloud, it reduces the need to store data of critical importance to the enterprise on mobile computers, thereby reducing the potential losses that may occur due to the loss of less critical data by an employee. This centralization also allows for greater ability to monitor data access, and to identify, increase and reduce authority when needed.

Friendly environment: Because a large number of servers are clustered in one location, they can be powered by a green card, such as solar panels or wind towers. The use of dedicated servers in their high rectangular form makes them operate less, and consume less power than the traditional large conventional server. At the moment, the United States and Europe are very concerned about the environment, encouraging individuals and companies to reduce energy and electricity use, and some expect them to start preferring companies that pursue policies that reduce energy use and carbon emissions.

2.4 Service Models

There are three main services of Cloud computing as shown in Figure 2.1,: Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS) [19].

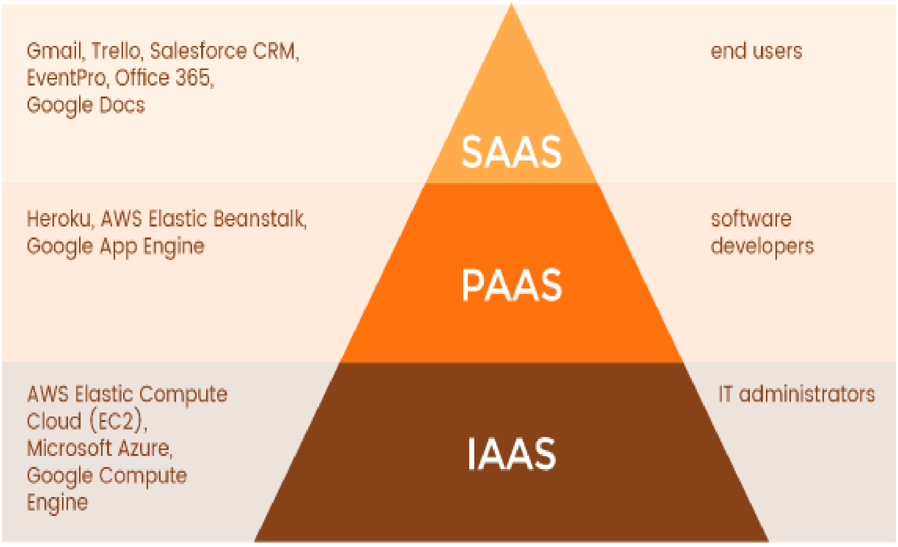

Gmail, Trello, Salesforce CRM, EventPro, Office 365, Google Docs
SAAS
end users
Heroku, AWS Elastic Beanstalk, Google App Engine
PAAS
software developers
AWS Elastic Compute Cloud (EC2), Microsoft Azure, Google Compute Engine
IAAS
IT administrators

Figure 2.1 Cloud computing service models [6]

2.4.1 Software as a Service (SaaS)

This model is provided by cloud service providers and aims to enhance the service quality of cloud and facilitate the computational tasks that may conduct over the cloud by the users. Cloud vendor may install such software into the cloud pool and made it available for cloud users. The practice of utilizing software as a service may happen against some fees charged by the vendor to the costumer. The term scalability may be used in here by the cloud consumers if they required to use such service on the cloud [17]. Price of those services can be adjustable and scalable. Cloud application such as Google Apps, online QuickBooks, video platforms such as Limelight, Emails, Office applications (e.g. Microsoft Office 365), file transfer platforms (e.g. FileShareT) are examples for SaaS. It is noteworthy to mention that cloud applications can be used (accessed) at anytime and anywhere globally with the help of the Internet [18].

2.4.2 Platform as a Service (PaaS)

Another alternative is innovated to provide access for many platforms and applications using the Internet without the need of actual purchasing of the same. A platform that can provide such a service is targeting the software developers and information technology professionals to develop some applications and post it over this platform and attract the individuals to use such services. Services of platform are provided relatively free for the developers where they can deploy their work to be used by platform pioneers. So many examples can be seen providing the same services like OrangeSpace, Mendix, Amazon Elastic Beanstalk, Microsoft azure and Google platform engine [19].

2.4.3 Infrastructure as a Service (IaaS)

Infrastructure as a service means the permission of the cloud granted to the customer to employ the cloud infrastructures such as hardware and software resources for any kind of legitimate operations. The clients of the cloud can enjoy services like CPU cycle, storage and memory operations, networking devices and equipment on the basis of renting without purchasing any of them and they can be billed against the actual utilized resources as per their requirements. Infrastructure as a services can be provided in either public cloud or in private cloud or in a hybrid fashion (mixing between private and public) [20]. Examples of infrastructure as a software are Amazon Elastic cloud computing, GoGrid, Linode, Terremark, Eucalyptus and cloud of RackSpace.

2.5 Deployment models

There are four forms through which Cloud systems can be deployed: private, public, community, hybrid Clouds [13].

2.5.1 Private Cloud

This model is applied only in organizations, and used exclusively by the employees of these organizations [21]. This model is controlled and administered by the organization itself or by a third party. In this model, the Cloud infrastructure is installed. Maintenance and administration will be easier in this deployment model. The level of safety is also high. Besides, the organization has more and more control over this model on the Cloud infrastructure, and has high accessibility [13, 22]. As shown in Figure 2.2, the organization is composed of the local environment and the cloud resources of the same organization are consumed by cloud users through an internal network of the company.

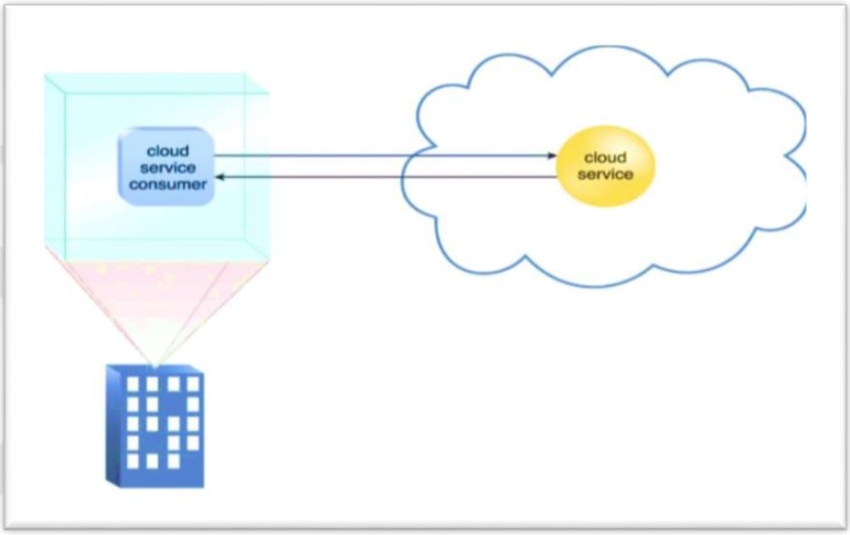


Figure 2.2 Private cloud [1, page 24]

2.5.2 Public Cloud

This model is applied to public users, managed and controlled by companies or organizations that sell Cloud services [21]. Also, users can get a Cloud depending on how long the Cloud is used. Because all applications and services are still available to all users, they are more vulnerable to security threats than the other Cloud models [22]. As shown in Figure 2.3, many enterprises are represented as consumers in the cloud while accessing cloud solutions hosted by various cloud service providers.

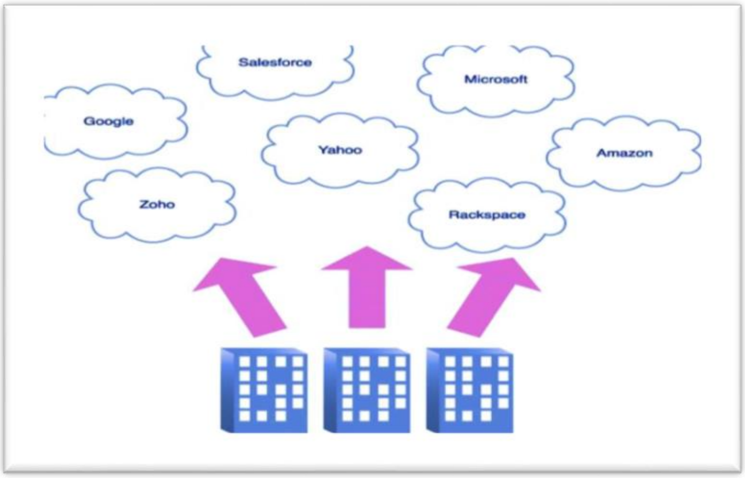


Figure 2.3 Public cloud [1, page 19]

2.5.3 Community Cloud

In this model many organizations are involved in the implementation of this Cloud model with common concerns (security requirements, policy considerations and missions) [21]. This Cloud is managed by one or more of the participating organizations, and may be managed by the third party [22]. Figure 2.4 illustrates a graphical representation of a community model of the cloud.

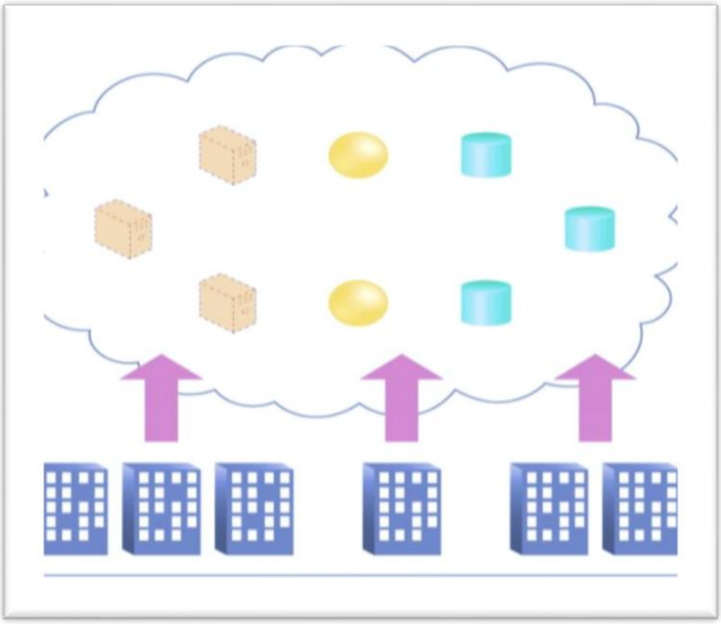


Figure 2.4 Community cloud [1, page 21]

2.5.4 Hybrid Cloud

The combination of two or more of the Clouds mentioned above (private, public, community) forms a hybrid cloud [22]. Clouds involved are connected with some standard protocols, helping the organization to meet its needs in the Cloud. As shown in Figure 2.5, consumed IT resources for an organization can be taken from both private and public clouds.

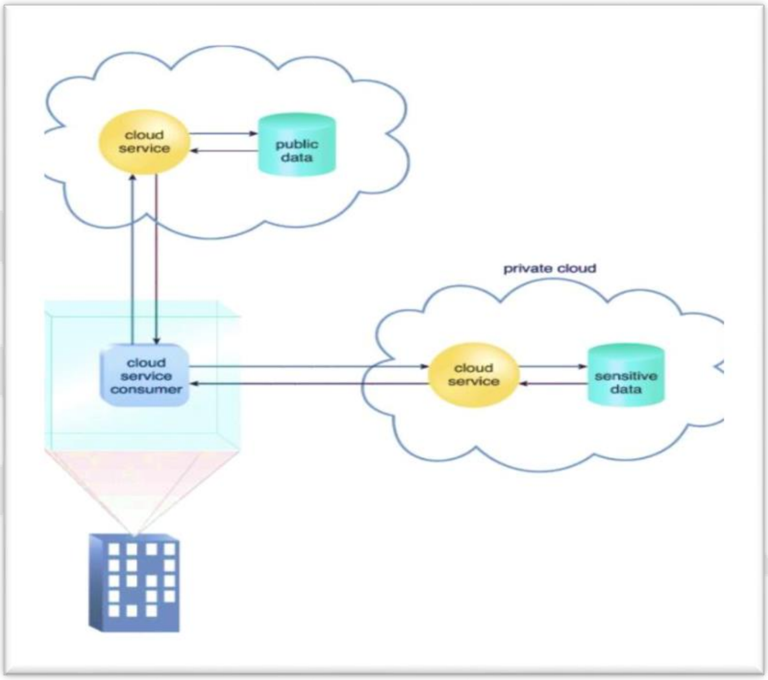


Figure 2.5 Hybrid cloud [1, page 22]

2.6 Scalability

Scalability is an essential approach in could computing, which intends to meet the demand of business uncertainty such as variation in data storage capacity (e.g. scale down or up the storage space due to business requirement fluctuation). With the concept of scalability, flexibility of cost and time are ensured, more likely, once the requirement of business is increased, server can be scaled up without complete transformation of the original infrastructures. Similarly, resources can be scaled down if business demands changed so that cost reduction is achieved without actually need to switch the service provider [23].

2.6.1 Types of Scalability

Scalability is one of the most important feature of Cloud computing. There are typically two type of scalability which are vertical scalability and horizontal scalability.

2.6.1.1 Vertical scalability

It also termed as up scaling which involves intervention of multiple resources with a particular entity so that they can work as united body. However, servers suffering from poor processing capabilities may adopt the vertical scalability to enhance the system performance. For example, system may claim of up scaling if the CPU speed of that system is not serving the purpose or even if the memory is smaller as compared to business demand. Vertical scalability can efficiently overcome the issues of internal process such as processing abilities of data, e.g. memory operations, CPU operations etc., more than its participation to enhance the storage capacity. Figure 2.6 can demonstrate the vertical clustering or up scaling of servers [24].

The disadvantage of this scalability is the latency increment which means, more processing power is required to process the large infrastructure (virtual machines) and this will not change in the vertical scalability so the time required to perform particular process is increased due to information size increment.

Figure 2.6 Clustering the servers in vertical scale [25, page 3]

2.6.1.2 Horizontal scalability

Horizontal scalability or out scaling is termed to the system capability to interface with several (as many as possible) software or hardware resources and work with them as one infrastructure (single logical entity). In other words, more units of the same kind of work can be merged together to perform bigger tasks. Reason of horizontal scalability may term to the requirement of extra computational capabilities such as processing speed or storage space which is achievable by adding more servers into the current server as per the requirements. So-to-say, one server may not meet the requirement of customers business so that out scalability or horizontal scalability is used to merge two, three, or ever more servers and allow them to perform the same tasks which can optimize the overall process [26]. Figure 2.7 shows clustering the servers in horizontal scale.

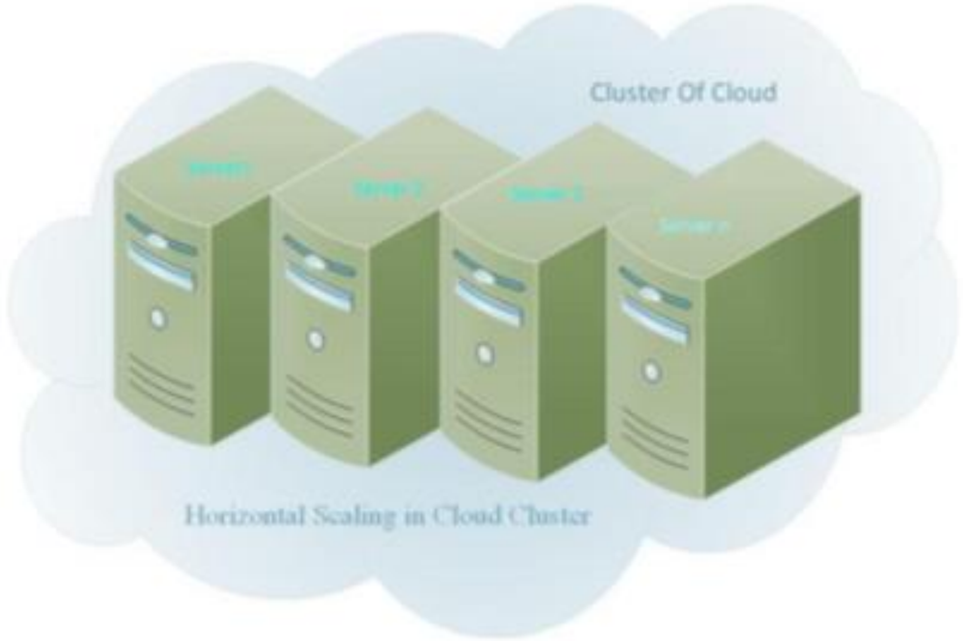


Figure 2.7 Clustering the servers in horizontal scale [25, page 3]

In this type of cloud scalability, additional nodes are added to the existing nodes for enhancement of processing (computational methods). Additional processing units may be applied to cope up with the requirements after scaling up the cloud. The main drawback of this method is the requirement of additional cost.

2.6.2 Auto Scaling

Auto Scaling is the ability to increase or reduce capacity automatically according to the user's specified conditions. With Auto Scaling, the number of instances increases seamlessly as demand increases to maintain performance, and automatically reduces as demand decreases to reduce costs [27]. Auto Scaling is implemented in many cloud service providers, for instance it is implemented in EC2 from Amazon. The minimum needs are a load balancer and two web servers. It needs set up an auto scaling algorithm and configure the threshold value based on the network traffic. When the threshold setting value is "PASSED", Amazon EC2 will rotate a new web server and automatically insert it into the load balancer assembly. Similarly, when the traffic falls below the threshold value, Amazon will take a server from the allocated pool [28].

Auto-scaling advantages [29]

- Auto scaling typically means allowing some servers to sleep during low load times, saving on electricity costs (as well as water costs if water is used to cool machines).
- Auto scaling can result in lower bills, because most cloud providers charge based on total usage rather than maximum capacity.
- Auto scaling can help by allowing the company to run less time-sensitive workloads on machines that get freed up by auto scaling during times of low traffic.
- Auto scaling solutions can also take care of replacing unhealthy instances and therefore protecting somewhat against hardware, network, and application failures, such as the one offered by Amazon Web Services.
- Auto scaling can offer greater uptime and more availability in cases where production workloads are variable and unpredictable.

Auto-scaling Techniques

Auto-scaling techniques can be based the following categories [30, 31]:

1. Static Threshold Based Rules (Rules)

2. Reinforcement Learning (RL)

3. Queuing Theory (QT)

4. Control Theory (CT)

5. Time-series Analysis (TS)

1- Threshold-based rules (rules)

Static threshold-based rules are typically used by cloud providers such as Amazon EC2. A simple example: if CPU > 70%, then scale out; if CPU < 30%, then scale in. It is quite difficult to set the correct thresholds and this must be done manually. An incorrect adjustment will cause oscillations in the number of VMs, and therefore, lead to bad performance. After each scaling decision, a cooldown period can be applied, during which no scaling will be performed. This cooldown period reduces the oscillations in the number of VMs and can be applied not only to rules but to any auto-scaling technique. In RightScale's [4] variation of rules, each VM votes independently, based on rules like those explained before, whether to scale or not. Then, simple democratic voting is performed to decide the scaling action.

2- Reinforcement learning (RL)

Auto-scaling based on reinforcement learning is a predictive approach to auto-scaling. Virtual machine instantiation is predicted via learned behavior. It makes decisions based on interaction between the auto-scaling agent and the scalable application. 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 remove resources to gain the maximum award (i.e., minimize response time). The main drawbacks of these approaches are bad initial performance, long training time and the problem to handle sudden bursts in input workload.

3- Queuing theory (QT)

Queuing theory can be used to add capacity by analyzing decisions and taking them based on a queue, for example, waiting requests in the load balancer. The classical queuing theory was widely used to model Internet applications and traditional servers to estimate performance metrics such as queue length or average latency. The approach is based on queuing theory, monitor of system parameters and application of specific performance laws (i.e., Little Law and Usage Law) to estimate system performance metrics. Since queuing theory only provides an estimate of performance metrics, most authors have combined it with other methods (i.e., threshold-based policies, control theory, and reinforcement learning) to deal with the problem of auto-scaling. There are two important obstacles to the use of queuing theory in auto-scaling systems. First, they impose unrealistic assumptions that are not valid in real scenarios; and secondly, they are not effective for complex systems.

4- Control theory (CT)

Control systems use a feedback loop by modifying the input of the control unit to influence standard outputs. The control systems are essentially interactive, but there are also some proactive estimates such as the typical predictive control, or even the combination of the control system and the predictive model. Control theory has been applied to automate resource management in various engineering fields, such as storage systems, data centers and cloud computing platforms. The main objective of the controller is to maintain the output of the target system (i.e., the performance of a cloud environment) to the desired level by adjusting the control input (i.e., number of VMs). Like line theory approaches, techniques based on the theory of control often use other supply methods (such as a threshold approach) to perform decision-making.

5- Time series analysis (TS)

Time series analysis includes a number of methods that use a past history window of a given performance metric in order to predict its future values (proactive techniques). In this case, we consider three methods: moving average, exponential smoothing and linear regression. Moving average calculates the mean of the n last values. Exponential smoothing assigns exponentially decreasing the weight to each value in the time series. Last, linear regression tries to fit a linear equation to the last values (where x is the time and y is the performance metric value), and then, it estimates a future value.

2.7 Scaling the Network

The process of resizing and replication on the virtual machine and database has made researchers think of the network which connects servers to each other. Networking over virtualized resources is typically done in two different manners: “Ethernet virtualization” and over-lay networks, and TCP/IP virtualization. These techniques are respectively focused in the usage of virtual local area network (VLAN) [36, 37]. The process of separating users' traffic is not sufficient to reach full application scalability. This leads to the need to scale the network in integrated datacenters that host multiple virtual machines per physical machine [38].

This scalability is often accomplished through over-provisioning resources to meet this increasing demand. In addition to expensiveness of this approach, it is static and does not take into account that not all the applications consume all the required bandwidth during all the time. Improved mechanisms taking into account actual network usage are required. On the one hand, one could periodically measure actual network usage per application and let applications momentarily use other applications’ allocated bandwidth. On the other hand, applications could request more bandwidth on demand over the same links [38]. Baldine et al. proposed to “instantiate” bandwidth-provisioned network resources together with the VMs composing the service across several cloud providers [39]. These authors employ Network Description Language (NDL)-based ontologies for expressing the required network characteristics. These abstract requirements are mapped to the concrete underlying network peculiarities [38].

These techniques to increase the utilization of the network by virtually “slicing” it have been dubbed as “network as a service”. By applying this mechanism, the actual bandwidth can be dynamically allocated to applications on-demand, which would benefit from a dynamic resource allocation scheme in which all the users pay for the actual bandwidth consumption. To optimize network usage statistical multiplexing is used to compute the final bandwidth allocated to each application [38].

Network slicing is a type of virtual networking architecture in the same family as software-defined networking (SDN) and network functions virtualization (NFV) — two closely related network virtualization technologies that are moving modern networks toward software-based automation. SDN and NFV allow far better network flexibility through the partitioning of network architectures into virtual elements. In essence, network slicing allows the creation of multiple virtual networks atop a shared physical infrastructure [38].

2.8 Load Balancing

Cloud computing like any other technology or engineering applications enables load balancing [31]. Load balancing can be defined as distribution of computational tasks among several servers which reduce the cost of adaptation of a new processing unit from the cloud. Load balancing is taking place at the cloud vendor premises where multiple applications are created and allotted for particular task, so, if requests arrived seeking same kind of task, request may get distribution among those application which may ensure delivering of powerful service in lesser time and minimal cost. Figure 2.8 depicts the process of load balancing wherein it shows users requesting service from the cloud and hence they can forward their request to the cloud using the web as usual and once request is arrived to the cloud server, server is processing the request through the load balancer which will segregate the service according to their type and forward them to particular servers (applications). This technology is called as software load balancing where several applications will serve single request. On the other hand, hardware load balancing is made to integrate several servers to attend large number of request so the load coming to one server is distributed among the others which may reduce the time and increase the efficiency of the service.

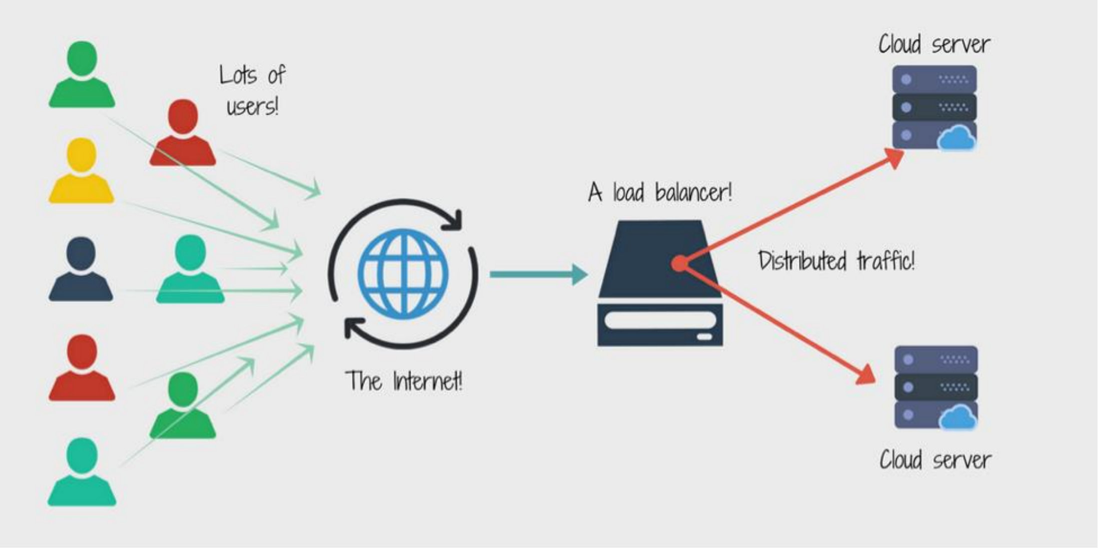


Figure 2.8 Cloud service load balancing strategy [7]

2.8.1 Load Balancing Technologies

Load balancing is the major solution to prevent the congestion of connection or service requests so that the total latency to deliver some service will be reduced. The herein are the most common techniques to perform load balancing [40].

2.8.1.1 Round Robin

This technique involves using of similar nature servers allotted to a same domain name. Those servers are performing similar tasks and hence they can receive a request related to those tasks only. The load distribution among those servers are taking place by assignment of IP addresses for all of those servers with same domain. A Domain Name System DNS server has a list of all the unique IP addresses that are associated with the Internet domain name. When requests for the IP Address associated with the Internet domain name are received, the addresses are returned in a rotating sequential manner.

2.8.1.2 Weighted Round Robin

This is supportive technique to the regular round robin where some servers are in sometimes going down so requests for those particular servers need to be reduced in order to meet the circumstances changing. In this case each server of the group is allotting a particular number (weight) and the server that required to process more requests are assigned to big weight number and that of small request processing is assigned a small weight number. Servers with higher ratings get more requests sent to them.

2.8.1.3 Least Connection

This is powerful solution for efficient service request distribution. It takes the request and forward it to the server which is experiencing a less load (less connections). So, it takes into account the current server load status while distributing the connections load.

2.8.1.4 Weighted Least Connection

It takes the similar concept of least connection to distribute the load among the servers. In a similar way alike round robin weighted, a numerical value is assigned to each server and server with higher value is receiving a maximum connection request whereas the server in lesser weight is receiving the minimum connection requests.

2.8.1.5 Agent-Based Adaptive Load Balancing

In this technique, each server in the pool has its own agent that monitors the current load of the server and reports it to load balancer. When making the decision to specify the best server to handle the requests, the real-time information in the reports provided by the agents is used to balance the load.

2.8.1.6 Chained Failover

Ordering servers in this technique will be in the form of a chain, the received request of connection is forwarded firstly to the first server in the chain then to the next server in the same chain and so on. In another word, if the request of connection is accepted then the request will be finished, else it will be forwarded to the next server of that chain.

2.8.1.7 Weighted Response Time

This technique aims to combat the congestion of connection requests by monitoring the respond times from all the servers and then determine which server can process the request faster. So, this information is used when next request arrives, which will be forwarded to the quicker server.

2.8.1.8 Source IP Hash

This technique is invented for maintaining the sessions of data transmission. Suppose server X and client Y are in process of exchanging particular information and the link between X and Y is suddenly goes down. Then, a unique identifier is generated to each client-server link and this identifier is being used later to resume the same connection after the link return.

2.9 Databases Scaling

The concept of database scaling is defined in the succeeding sections where database can be scaled up or scaled down to meet the demand of business requirement. Database is scaled up by adding additional units such as RAM and CPU (resizing) so that the resulting database is more reliable and efficient. Databases are available in several types such as SQL, PostgreSQL, Oracle etc. [32].

2.9.1 Replication of database

This process is made to maintain the continuity of a service, in other words, continuous availability of database by creating additional copy of the database. The data can be copied from the original server into other server and hence if the original server goes down for any particular reason, data will be still accessible. Databases are usually available in the cloud from different vendors and database scalability is one of the key points provided by the cloud for the users if they wished to expand the database at any time.

2.9.1.1 Master Slave Replication

It is a strategy of data replication for tackling some problems related to database security, system performance and sudden failure of systems. The database is saved elsewhere and can be accessed in any failure case. This technique is used to form several slaves (servers) and connect them to one main server (the original database server). The master is the original database server which is linked to multiple severs or single server as shown in Figure 2.9. The master server sends data to those slave servers as an updates and data replication of master slave is to be done in synchronous or asynchronous fashion. In synchronous replication, master may send update to the slaves and slaves are implementing the update at the same time and acknowledging that update is implemented. In other case, asynchronous replication means that server (master) is scheduling the update and implementing it at the slaves servers in some other time [33].

In single master - multiple slaves' replication, different type of databases can be participating the replication process such as:

- Replication from Oracle database to PostgreSQL database
- Advance server PostgreSQL Plus and Oracle replication
- PostgreSQL and SQL Server replication

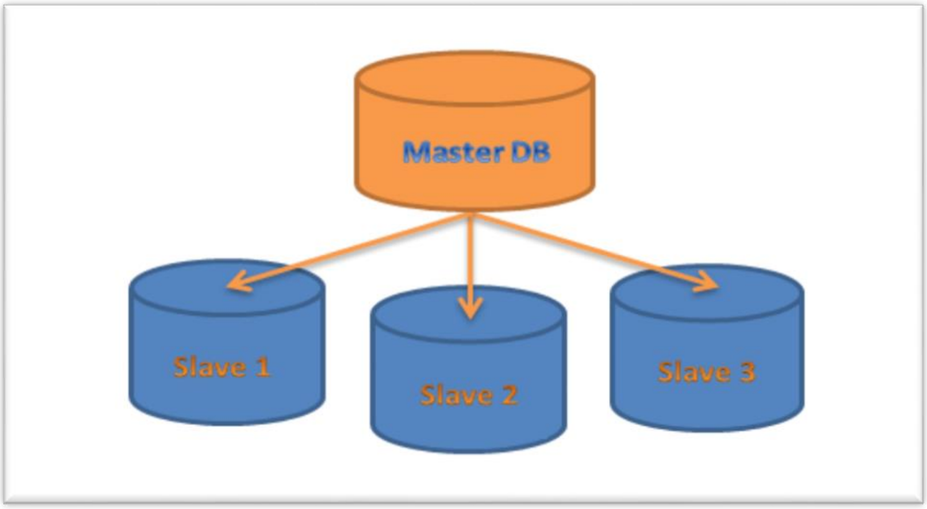


Figure 2.9 Master slave mechanism of data replication [2]

2.9.1.2 Multiple Master Replication

Multiple Master Replication is done when multiple databases are made of master nodes and enables databases to be updated and reflected in all databases in the other nodes. The Update can be insertion of new data into particular table in the database or deleting some particular table content or updating particular table content. In other words, any updating or change at particular node tables will be replicated to all other nodes tables [33]. Figure 2.10 is demonstrating the concept of multi master replication.

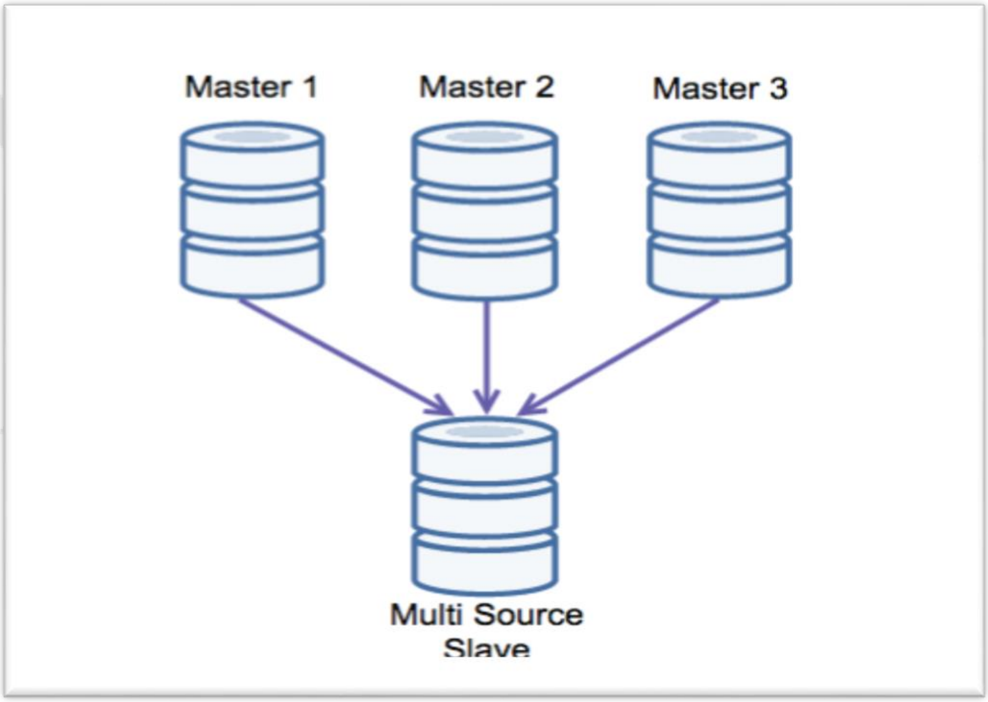


Figure 2.10 The multiple master replication in database server [5]

In multiple master replication, the type of databases has to be identical in order to participate the replication process such as:

- Replication from Oracle database to Oracle database
- Replication form PostgreSQL database to another Postgre plus advance server database
- Advance server PostgreSQL Plus and itself
- PostgreSQL Servers replication

2.9.2 Database Partitioning

It a technique to simplify the big tables in databases into smaller handy tables which helps to control or problems that usually occur in databases. The partitioning technology is taking place by using the portioning key. Partitioning key is a matrix of number of columns and rows that is used to define the boundaries of database partitioning [34].

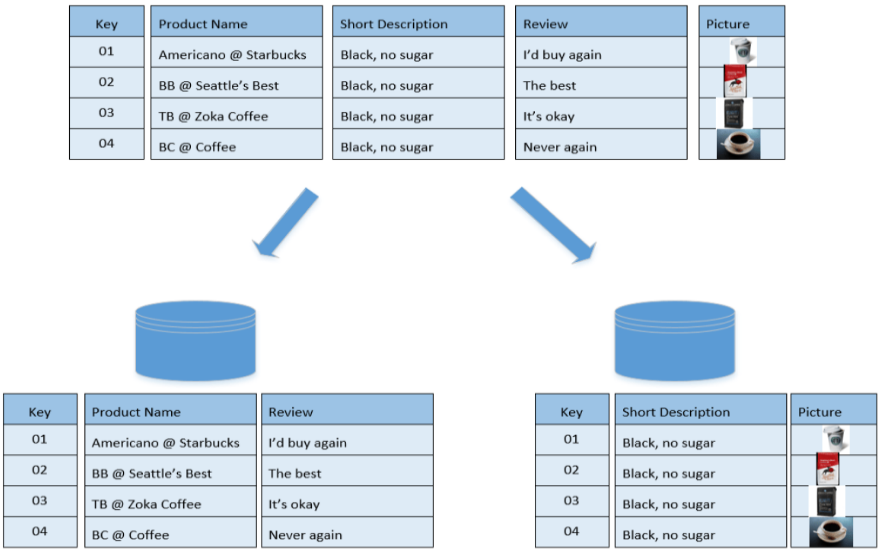
2.9.2.1 Strategies of Partitioning

The first strategy of partitioning is the range strategy which is the most popular strategy on data portioning and take place by using the ranges in the key partitioning as partition boundaries. The partition is associated with less than clause value to specify the upper limit of partition which is not included in the data. For mapping the data into particular partition, some databases alike Oracle is deploying Hashing algorithms for that purpose. In hash partitioning, the resultant tables will hold similar size approximately as the algorithms distributing the row data evenly among the partitions. This technique (hashing) is the best for supplying the data across the several devices. The next partitioning technique is called as “List partitioning”, this technique is invented to group the data that not having particular order or data that is not relative. Another portioning techniques are also defined such as interval partitioning and virtual column based which are respectively defined as: partitioning on the bases of interval definition of table rows and by using extra column as virtual key portioning [35].

2.9.2.2 Vertical and Horizontal Partitioning

Portioning the big database table in vertical fashion or in horizontal fashion is important to be adopted in databases. Vertical partitioning is involved neglecting of particular information in the table of data due to the insistence of demanding only small table or small part of data. The vertical partitioning can be taken place by either referring the index along with particular costumer in the table such as the index and costumer name or index and most sold product or index and costumer address. Figure 2.11 is depicting this technology.

In horizontal partitioning of data, particular rows can be referred as list of particular names of costumers with their most sold product and address and contact number. Figure 2.12 is depicting this strategy.

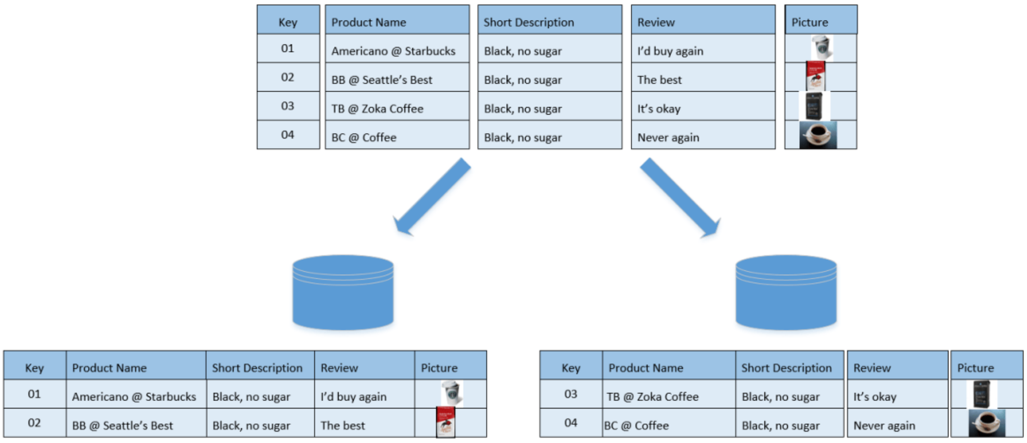


Key	Product Name	Short Description	Review	Picture
01	Americano @ Starbucks	Black, no sugar	I'd buy again	
02	BB @ Seattle's Best	Black, no sugar	The best	
03	TB @ Zoka Coffee	Black, no sugar	It's okay	
04	BC @ Coffee	Black, no sugar	Never again	

Key	Product Name	Review
01	Americano @ Starbucks	I'd buy again
02	BB @ Seattle's Best	The best
03	TB @ Zoka Coffee	It's okay
04	BC @ Coffee	Never again

Key	Short Description	Picture
01	Black, no sugar	
02	Black, no sugar	
03	Black, no sugar	
04	Black, no sugar	

Figure 2.11 Vertical portioning of database tables [4]



Key	Product Name	Short Description	Review	Picture
01	Americano @ Starbucks	Black, no sugar	I'd buy again	
02	BB @ Seattle's Best	Black, no sugar	The best	
03	TB @ Zoka Coffee	Black, no sugar	It's okay	
04	BC @ Coffee	Black, no sugar	Never again	

Key	Product Name	Short Description	Review	Picture
01	Americano @ Starbucks	Black, no sugar	I'd buy again	
02	BB @ Seattle's Best	Black, no sugar	The best	

Key	Product Name	Short Description	Review	Picture
03	TB @ Zoka Coffee	Black, no sugar	It's okay	
04	BC @ Coffee	Black, no sugar	Never again	

Figure 2.12 Horizontal portioning of database tables [4]

2.10 Scaling Virtual Machines

Virtual machines are nothing but computers that are virtually created and used to provide computing service over particular cloud. The popular virtual machines are used as part of physical computers to run additional operating system and perform the other tasks of normal machine. In the context of cloud computing, virtual machine is used as platform to run particular software on that cloud environments. Virtual machines can be scaled as one of cloud units. Cloud is usually including large number of virtual machines in the range of hundreds, these virtual machines can be scaled up automatically using the virtual machine scale set. Big cloud vendors such as Microsoft Azure are providing the tool to shrink or expand the virtual machine number depending on the requirement in automatic way. The other cloud vendors are doing the same, a set scale of virtual machine is the only helping tool to scale down or up the virtual machine automatically without thinking of the networking (how to link the additional machines to the original ones) or other setups, it may require to change simple properties on the cloud toolbox [35]. Figure 2.13 is depicting the process of nodes scaling in cloud.

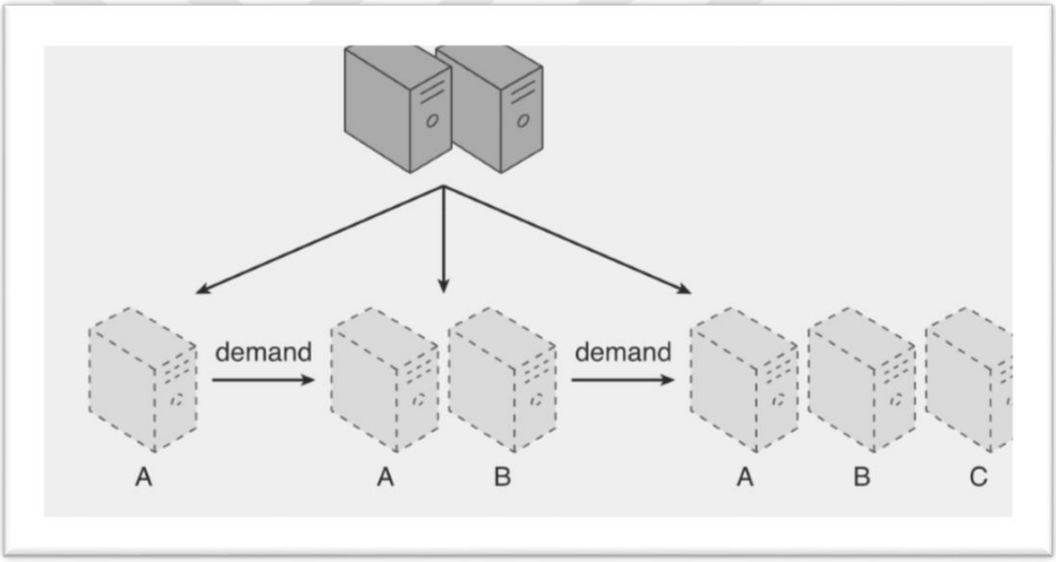


Figure 2.13 Node scaling in cloud [3]

CHAPTER 3

LITERATURE SURVEY

3.1 Literature Survey about Systematic Mapping

Systematic mapping is a specific statistical method for forming a classification scheme that utilizes information about research questions by analyzing the results of the frequencies of publication over time, thus understanding the scope of the research coverage. It is also possible to combine different aspects of the results to answer more specific research questions [41].

A systematic mapping study provides a clear visual representation of a particular scope of work, making it a useful tool for researchers and beneficiaries. Many researchers have addressed systematic mapping in the fields of computer science.

Ozkoc and Ogutcu [42] conducted a systematic mapping of the trends of technology transfer studies for a period of five years. They divided the study into technology transfer, transfer of technology, and technology transfer efficiency. In addition, the total number of studies in this research was 59 studies.

The paper presented by Fleh et al. [43] provides a systematic mapping study on the recognition of the social touch gesture. 49 research papers were selected from a total of 938 papers collected from various sources. These papers were selected between 1996 and 2017. The study showed that most of the research was published in 2015, most of those were published in conferences, and placed in the evaluation papers and verification papers.

Tjong et al. [44] found a framework for the project engineering work carried out in Indonesian higher education. The higher education institution needs for engineers, projects and the implications of their adoption were studied. Within 685 papers, 44 papers were approved as candidates, and finally 12 papers were selected. The results of the research concluded that most institutions of higher education were willing to improve their work quality under the influences of the institution's policy.

Justo et al. [45] recognized that by identifying the software designs in the requirements engineering phase of the project, the developing phase of the systematic mapping study was discovered. Based on the basic criteria required in the development process, the roles played by the developing stage were understood. The study concluded that to enable replication of these works by the research community, a study protocol was developed with the basic steps of the study to validate the research.

Kosar et al. [46] described the systematic map protocol for the period July 2013 to October 2014 with respect to domain-specific languages (DSL). The study gave proper attention to subsequent expectations, research trends, actions on domain-specific languages (DSL), and requirements for a systematic review.

Santos et al. [47] created a systematic mapping by using concept maps in computer science. The study focused on supporting learning, teaching, collecting, and analyzing the previous articles on the concept maps, which led to a full examination of concept maps. The searching process involved the utilization of backward snowballing and manual methods. The search strings are applied on SCOPUS, Science Direct, Compendex, ACM DL, and IEEE Explore digital library sources.

Based on publications for the period between 1974 to 2016, Souza et al. [48] considered 156 primary studies in their research about the methods of games engineering and how these methods improve engineers. The study also focused on the uses of games in the teaching of engineering software.

Ahmad et al. [53] consider 69 out of 75 research papers in their study for accurate statistical analysis of the cloud-based tests in the process of constructing a classification scheme. The quantitative result was determined. The study examined functional and non-functional test techniques; these applications properties and corresponding tools were also used to predict future results.

The study presented by Ayo et al. [49] provided six classes of studies in the fields of architecture, virtualization, application, improvement design, implementation and presentations related to the study scope. A systematic mapping study was presented for high-performance computing and cloud. Selected studies have been applied to the contribution of this method, tool and model. The selected studies were used on the research side, which deals with assessment, verification and research solutions.

Ayo et al. [50] examined the systematic mapping of cloud resource management and the scalability of scheduling, capacity planning, brokering and flexibility. These key features were discussed in the resource management classification scheme. Selected studies have been applied to research types and to the contribution of them, such as measurement, instrument, method, and modeling.

Nine research questions were put forward by Alayyoub et al. [51] and 91 studies were considered for the period between 2010 and 2015. The study examined the trends and differences on several effective stream processing frameworks (SPFs) by categorizing research on SPFs and regional scales that the researcher might consider to get an overview of this field.

3.2 Literature Survey about Cloud Scalability

Although there are a lot of studies related to scalability in literature, there are only a limited number of studies about scalability in cloud computing. One of these studies is done by Liu et al. [52]. They discuss scalability types for both vertical and horizontal scalability in cloud computing. Moreover, authors explain the advantages and disadvantages of each type. Also, they made an experimental test to show the difference between them in performance and to show the better one in different cases. Another study is carried out by Sotiriadis and his colleagues [53]. They discuss vertical and horizontal scalability. They focus on the load-balancer on virtual machine when there is a variation on load from different users. They made an experimental study for both vertical and horizontal scalability and compared them.

Hwang et al. [54] present three kinds of scaling techniques, namely scale-up, scale-out and auto-scale. In this study, the authors evaluate these three strategies by using different criteria such as performance, efficiency and productivity. There is another study done by Alipour et al. [55], carried out a survey on the concept of auto-scaling techniques. Based on the result of the survey, they show the main issues that affect auto-scaling in cloud computing. They provide a direction for the researchers in various areas in auto-scaling. They presented deployment models, scalability dimensions and service models. The authors mentioned that cloud can be scaled up or down vertically or scaled out or in horizontally, and hence, they descripted the concept of automatic scalability and the challenges faced when auto-scaling is used.

Falatah et al. [56] give a brief definition of scalability in cloud which has the ability to do the works given by users in a fast manner. There are some criteria that you should pay attention such as load balancing, resource allocation, and optimization. Also, the authors, in this study, illustrate the performance of scalability and its levels. Moreover, authors then show the approaches of scalability and they give details about web application in the cloud. Likewise, M.Kriushanth et al. [27], shows concepts of cloud computing such as deployment models, services model and types of scalability. Also, they focused on vertical and horizontal scalability in cloud computing. Finally, they discuss the infrastructure and the main issues and challenges that impact on auto-scaling.

Hung et al. [57] introduced an algorithm based on auto-scaling for automated provisioning and balancing of virtual machine resources for active application sessions. The algorithm takes into consideration the energy cost. Moreover, the algorithm has the ability to handle sudden load requirements and maintain higher resource utilization. Finally, authors proposed two kinds of algorithms for distributed systems and for auto-scaling of web applications.

Trieu et al.[58] introduced a new algorithm for automated provisioning of virtual machine resources depending on the number of thresholds of the active sessions. Furthermore, the ability of cloud is discussed upon request to provide resources and dynamically allocate them to users. In addition, the purposed algorithm shows the ability to handle some sudden load surges, maintaining higher resource utilization, and delivering IT resources on-demands to users. Thereby, it results in reducing infrastructure and management costs.

Nandgaonkar and Raut [59] give a comprehensive study on cloud computing technology. Cloud has some services provided to users in a cloud environment with flexibility, scalability and so on. Plus, they explain the privacy, security, and internet dependency and availability as avoidance issues. The authors discussed the scalability techniques and the challenges for both horizontal and vertical scalability.

Mickulicz et al. [40] mentioned that cloud base software services require to be observed from the performance point of view. Thus, the testing and measuring of cloud performance to be performed frequently due to the rapid expansion of the cloud service in terms of scalability and elasticity. Also, the capacity of a cloud can be increased in two ways: service volume expansion — when single instance used for serving the requirements - or by increasing the number of instances themselves, more likely, using multiple instances to provide the service instead of a single instance. Finally, authors focused on the average time of response as a performance metric of cloud when the instance is increased in volume or in number.

Lorido-Botrán et al. [31] presented that auto-scaling can work in a direct reactive way which means servers can be scaled out or scaled in as soon as the demands of resources decrease or increase. Moreover, a proactive auto-scaling can be used as a predictor of demand fluctuation, which means system administrations may adopt an algorithm to determine the future demand and accordingly make a plan of auto-scaling. The authors explain several algorithms that can perform a proactive approach of auto-scaling.

In the same way, Iqbal et al. [60] mentioned that scalability can be recognized by its parameters. Four parameters are used to recognize scalability: processor, input/output throughput, configuration of server and storage capacity. They stated that scalability deficiency can be tackled by using the concept of auto-scalability which involves automatic adjustment and scaling (distribution) of cloud resources while maintaining the minimum level of performance and cost. Auto scalability can be achieved using some software and applications such as Amazon Web Services (AWS). Auto scalability may not only upscale or downscale the servers, but it can also maintain cost efficiency.

CHAPTER 4

METHOD

4.1 Research methodology

In this study, in order to conduct a systematic mapping study, the steps mentioned in [61] are followed. The proposed study begins by defining the research questions, then performs a number of systematic maps, which are the answers to the research questions as shown in Figure 4.1.

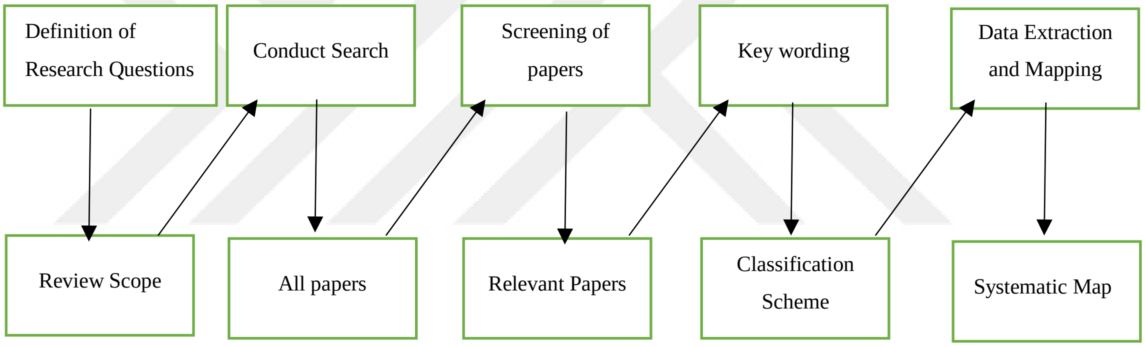


Figure 4.1 Process steps and outcomes

The main objectives of systematic mapping studies are to identify and quantify the research in its area, predict the quality and direction of the research, and set the publishing frequencies with the time and forums in which the research was published, thus providing an overview of the research field.

Papers were collected from available databases using the search strings of the subject in question, then a screening process was conducted for the collected papers using the inclusion and exclusion criteria. The papers are mainly classified according to the type of scalability; vertical and horizontal.

4.1.1 Research questions

The following research questions were identified as relevant to the purpose of this study:

RQ1- Which database includes the most relevant studies related to horizontal and vertical cloud scalabilities?

RQ2- What type of venue (conference/journal) has the most published papers for each corresponding year?

RQ3- What are the methods used or proposed in vertical scalability in cloud computing? And the corresponding annual number of papers published?

RQ4- What are the methods used or proposed in horizontal scalability in cloud computing? And the corresponding annual number of papers published?

RQ5- Which of the proposed methods are used by cloud service providers?

RQ6- Which type of cloud scalability is receiving the greatest focus of studies?

RQ7- What type of research is conducted on cloud scalability?

4.1.2 Search sampling

In the current study, the digital libraries listed in Table 4.1 have been accredited for the purpose of collecting papers for their international reputation and their widespread dissemination of research in journals and the proceedings of the world scientific conferences. Each digital library was used several times in order to be sure that the study covered all the relevant papers from these sources.

Table 4.1 Databases used in the systematic review

Database	Sources
IEEE Explore	http://ieeexplore.ieee.org/Xplore/home.jsp
Science Direct	http://www.sciencedirect.com/
ACM Digital Library	http://dl.acm.org/
Google scholar	https://scholar.google.com.tr/

A search string was used based on the quality of the research required. The search string used to describe the subject of the study to be studied was as follows:

“Cloud Computing “AND” horizontal scalability” OR “vertical scalability”

4.1.3 Screening the Papers

A large number of scientific papers were collected for the subject. The collection process was completed in May 2019. The papers were screened according to the inclusion and exclusion criteria given in Table 4.2 and Table 4.3. Then the papers were sorted according to the main title of the research, the summary, and the keywords mentioned in the papers. After completing the first screening, the second screening process was done by reading the papers in full to exclude according to the criteria mentioned in Table 4.2 and Table 4.3.

Table 4.2 Inclusion criteria

SI	Inclusion criteria
1	Studies including vertical scalability in cloud computing.
2	Studies including horizontal scalability in cloud computing.
3	Journal or conference papers.

Table 4.3 Exclusion criteria list

SI	Exclusion criteria	No. of Studies
1	Studies that do not accessible in full text.	15
2	Studies that do not address the horizontal or vertical scalability.	732
3	Studies that do not presented in English.	42
4	Prefaces, slides, panels, editorials or tutorials.	50
5	Studies that do not answer the research questions.	313
6	Studies that are duplicated among the databases.	53

4.1.4 Keywording

"Keywording" is an effective tool for reducing the time needed to design a classification scheme to manage cloud computing model studies. It also ensures that the important papers are taken into account in the scheme. Abstracts and conclusions were studied for the extraction of the keywords related to this study. Therefore, the keywords were combined to give satisfactory knowledge about the kind of contribution research. This was eventually used to determine the facts or categories of the study. A set of keywords was used to identify the categories and the final systematic map. Finally, we identify two areas of horizontal and vertical scaling which are; *virtual machine or container* and *database* in addition to proposed methods that used for scaling such as; replication, partitioning and resizing.

4.1.5 Research type

In order for the study to be comprehensive, the research methods were classified into the following types [62]:

1- **Validation papers:** Techniques investigated are novel and have not yet been implemented in practice. Techniques used are for example experiments, i.e., work done in the lab.

2- **Evaluation papers:** Techniques are implemented in practice and an evaluation of the technique is conducted. That means, it is shown how the technique is implemented in practice (solution implementation) and what are the consequences of the implementation in terms of benefits and drawbacks (implementation evaluation).

3- **Philosophical papers:** Techniques that provide new methods to solve a problem related to the framework and concepts are discussed.

4- **Opinion papers:** These papers express the personal opinion of somebody whether a certain technique is good or bad, or how things should be done. They do not rely on related work and research methodologies.

5- **Experience Papers:** This paper provides a look at how to do something through the personal experience of the researcher.

The above categories were adopted in the classification scheme of this study and considered sufficient. All papers in this study were examined on the basis of different research categories.

4.2 Results and Discussion

The purpose of the current study is to determine the frequency of publication in each category, and to identify the category that is most discussed by researchers in the previous research from the analysis. The gaps were identified using the systematic map, which shows the subject areas that were addressed. The answers for various research questions were sorted in forms of tables and graphs.

According to the information obtained during the study, we note that the study period, which extends from 2006 to 2018, can be divided into two periods. The first period is from 2006 to 2012, where the publication of papers on the cloud scalability was very limited. It can be referred to as “The poor period”. The second period is from 2013 to 2018, which was characterized by a large number of papers published on cloud scalability, may be referred to as “The rich period”.

RQ1- Which database includes the most relevant studies related to horizontal and vertical cloud scalabilities?

The number of papers after the screening, which will be covered by the current study, is shown in Table 4.4. Figure 4.2 shows that most of the papers were collected from IEEE explore (81 papers, 41 %) followed by ACM Digital Library (47 papers, 24 %), Science Direct (44 papers 22 %) and Google Scholar (26 papers, 13 %).

Table 4.4 Number of papers before and after screening

Database	No. of papers before screening	No. of papers after screening
IEEE Explore	329	81
Science Direct	319	44
ACM Digital Library	217	47
Google scholar	340	26

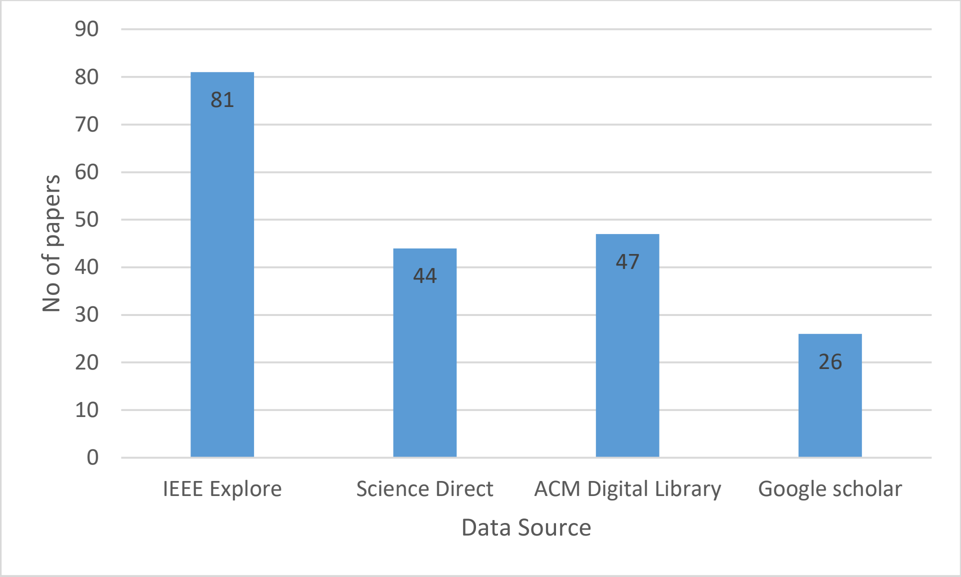


Figure 4.2 Number of papers according to data sources

Table 4.5 Annual publication count for scalability

Year of publication	Horizontal scalability		Vertical scalability		Total
	Journal	Conference	Journal	Conference	
2006	1	1	-	-	2
2007	1	1	-	1	3
2008	2	1	-	-	3
2009	2	2	1	3	8
2010	-	5	-	2	7
2011	3	8	1	5	17
2012	4	13	3	4	24
2013	5	7	1	1	14
2014	5	11	2	4	22
2015	10	9	3	1	23
2016	6	8	5	5	24
2017	15	12	1	2	30
2018	10	5	5	1	21
Total	64	83	22	29	198

RQ2- What type of venue (conference/journal) has the most published papers for each corresponding year?

The papers were mapped to the type of venue (conference/journal) as illustrated in Table 4.5 and Figure 4.3 compares the studies covered between the years 2006 and 2018. Conferences had the largest share of papers. On the subject of horizontal scalability, there were 83 papers out of 147, and on the subject of vertical scalability, there were 29 papers out of 51. 56.5% of the papers were published through the proceedings of scientific conferences compared to 43.5% having been published in scientific journals.

Figure 4.4 shows the annual distribution of published papers for horizontal scalability between the years 2006 and 2018. The figure shows that 2017 was the year having the maximum publications about the horizontal scalability followed by 2015. In last fewyears the papers published in scientific journals is less than the proceedings of the conferences, as like as most other years.

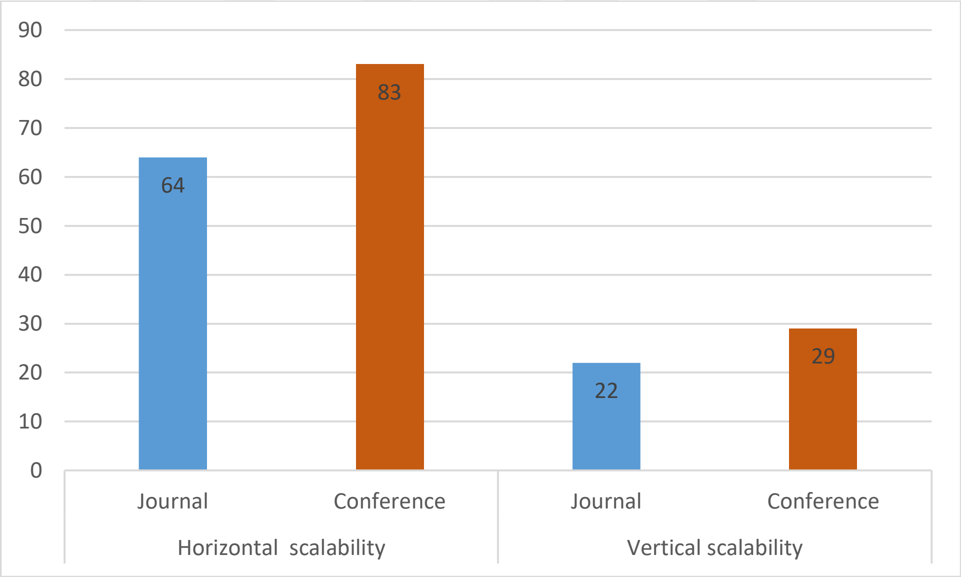


Figure 4.3 Number of papers for horizontal and vertical scalability

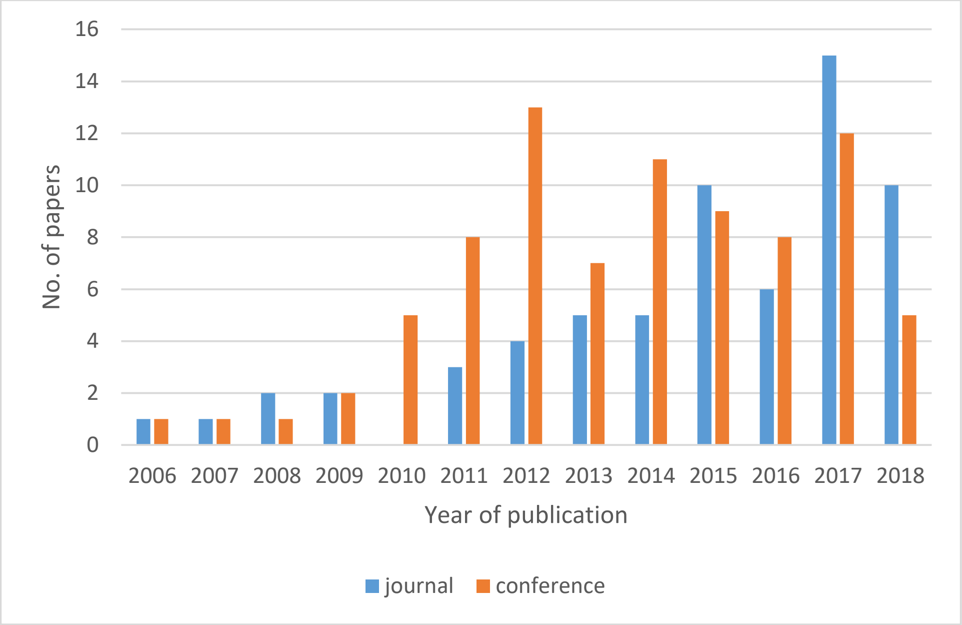


Figure 4.4 Annual publication count for horizontal scalability

Figure 4.5 shows the annual distribution of vertical scalability papers between 2006 and 2018. It is clear here that the year 2016 had the most published papers on vertical scalability.

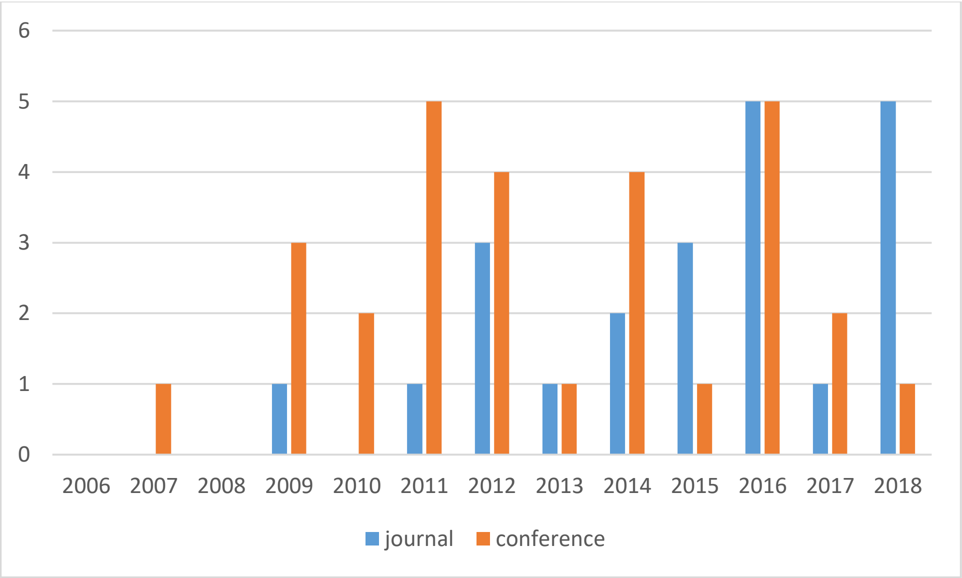


Figure 4.5 Annual publication count for vertical scalability

RQ3- What are the methods used or proposed for vertical scalability in cloud computing?

Table 4.6 and Figure 4.6 show the annual distribution of studies for vertical scalability related to *virtual machine or container*, *network*, *auto-scaling* and *database*. It is clear that in the period between 2006 and 2018, 17 papers on the subject of virtual machines were published, while 9 were published on the subject of database, 19 were on the subject of auto-scaling and 9 papers for network in vertical scalability.

The method of vertical scalability of virtual machines was a resizing method, and the same type of method was used for database and resizing bandwidth was used for network in vertical scalability.

Figure 4.7 shows the percentage of each type of vertical scalability published between 2006 and 2018. It is clear that the highest percentage was 35.5% for auto-scaling followed by the virtual machine (resizing) 31.5%, database (resizing) and network (resizing bandwidth) 16.5% for each.

Table 4.6 and Figure 4.6 show that most of the papers were published between 2014 and 2017. In reference to Figure 4.6 and Table 4.6, corresponding to the year 2014 and 2017, three papers were issued for virtual machine resizing, while in 2016 six papers for virtual machine resizing were published and two papers were issued for database resizing. In the year 2017, three virtual machine resizing papers were published and no papers were issued for database resizing.

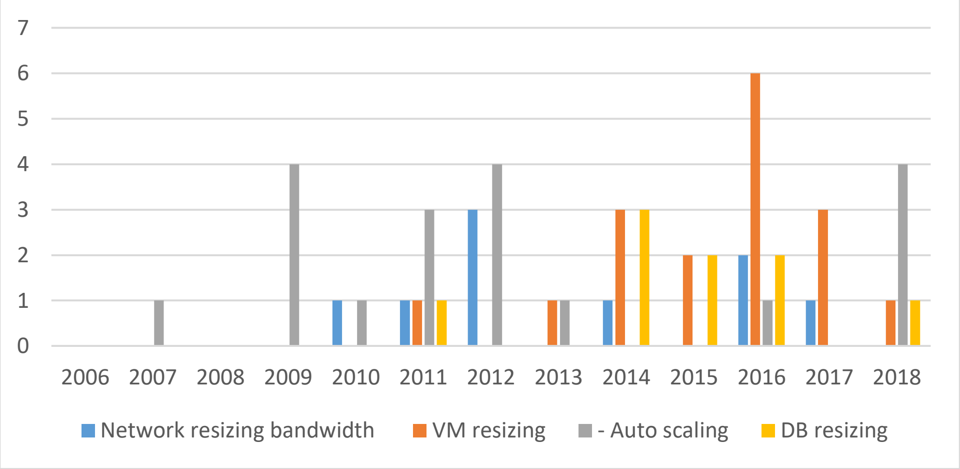


Figure 4.6 Annual publication count for vertical scalability types

Table 4.6 Annual publication count for vertical scalability types

Year of publication	Vertical Scalability			
	Network	virtual machine or container	Auto scaling	database
	resizing bandwidth	resizing	Auto scaling	resizing
2006	0	0	0	0
2007	0	0	1	0
2008	0	0	0	0
2009	0	0	4	0
2010	1	0	1	0
2011	1	1	3	1
2012	3	0	4	0
2013	0	1	1	0
2014	1	3	0	3
2015	0	2	0	2
2016	2	6	1	2
2017	1	3	0	0
2018	0	1	4	1
Total	9	17	19	9

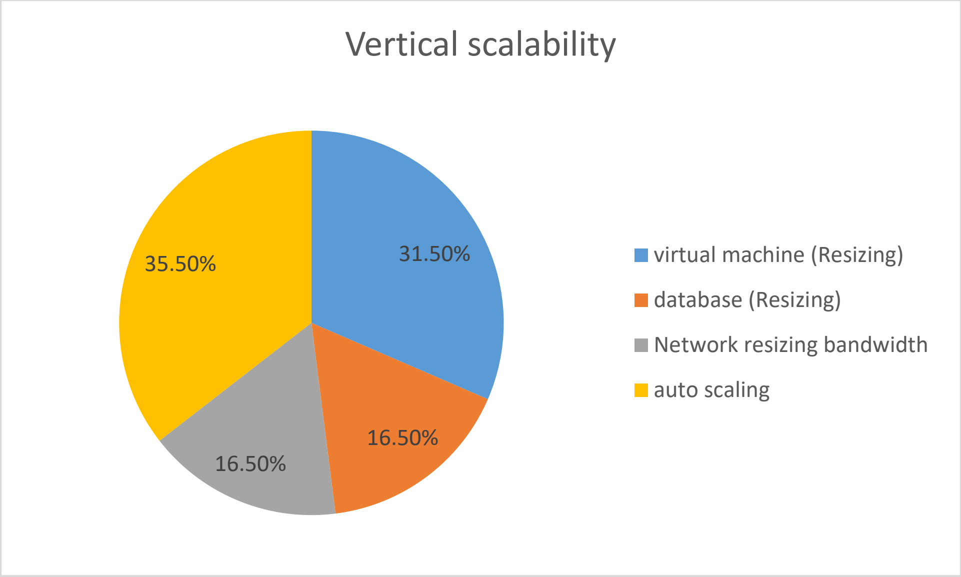


Figure 4.7 Percentage of publication count for vertical scalability

RQ4- What are the methods used or proposed in horizontal scalability in cloud computing?

Like vertical scalability, Table 4.7 and Figure 4.8 show the annual distribution of studies for horizontal scalability related to *virtual machine or container*, *network*, *auto-scaling* and *database*.

The database was divided into two types: the first is replication, and the second is partitioning, and we noted that the number of papers published for replication and partitioning were 34 and 24, respectively, also for auto-scaling the number of papers was 40. Figure 4.9 shows the percentage of each type of horizontal scalability published between 2006 and 2018. It is clear that the highest percentage was recorded as 26% for auto-scaling, 22% for the database (partitioning) and 16% for database (replication), followed by the virtual machine (replication) by 22%, then 15% for network slicing.

Table 4.7 Annual publication count for horizontal scalability

Year of publication	Horizontal Scalability				
	Virtual machine or container	Database		Network	Auto scaling
	Replication	Replication	Partitioning	slicing	Auto scaling
2006	0	0	1	0	1
2007	0	0	1	0	1
2008	0	0	0	1	2
2009	1	1	1	1	3
2010	0	0	0	0	5
2011	1	0	1	1	8
2012	3	2	0	3	9
2013	2	5	4	0	2
2014	3	5	4	2	1
2015	6	5	6	1	1
2016	6	4	1	2	3
2017	10	8	3	6	1
2018	2	4	2	5	3
Total	34	34	24	22	40

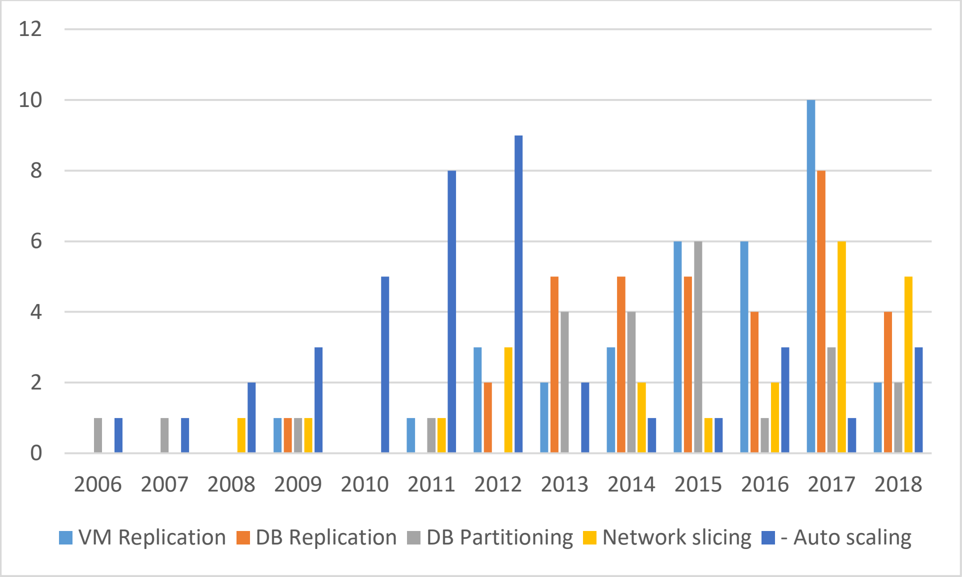


Figure 4.8 Annual publication table for horizontal scalability types

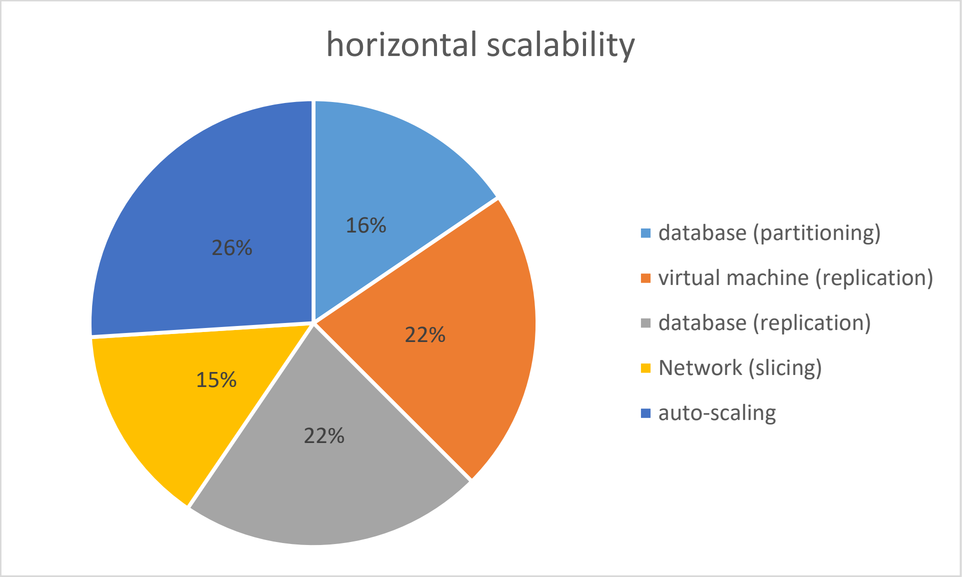


Figure 4.9 Percentage of publication count for horizontal scalability

The table and the figure show that most of the papers were published between 2015 and 2017 (quite similar to vertical scalability). In reference to Figure 4.8 and Table 4.7, in 2015, six papers were issued for each of the virtual machines (replication) and database (partitioning), while five papers were published for the database (replication). In 2017, ten replications of virtual machine papers were published, eight papers for database replication and three for partitioning.

RQ5- Which of the proposed methods are used by cloud service providers?

The cloud is a large scale solution on which Cloud Service Providers (CSPs) (e.g., Google, Microsoft, Amazon) are vendors who lease to their customers cloud services that are dynamically utilized based on customer's demand [63].

Amazon.com [64] is one of the most popular CSPs, which uses replication method of virtual machine or container and database, in horizontal scaling in addition to partitioning methods of the database. However, in vertical scaling Amazon uses resizing methods by adding resources or upgrading the existing capacity to a bigger one [65-67].

Microsoft Windows Azure is a platform on cloud that offers various types of services such as Web development, Mobile development platform, Storage and so on [68]. Like Amazon, Microsoft also uses replication method of virtual machine or container and database, in horizontal scaling and partitioning methods of databases [69-71]. Thus, in vertical scaling Microsoft uses resizing methods by adding resources or upgrading them [72]. Moreover, Google [73] use replication [74, 75], partitioning [76] and resizing [77] scaling methods as Microsoft and Amazon use.

RQ6- Which type of cloud scalability is receiving the greatest focus of studies?

Figure 4.10, Tables 4.6 and Tables 4.7 show the number of published papers for each type of cloud scalability within the studied period (2006-2018). The study clearly demonstrates that horizontal scalability was the highest of the published papers (149 papers) while there were 49 papers on the subject of vertical scalability, during the same period. The auto-scaling in horizontal scalability was the most popular publication subject, where the number of papers published is 40 for auto-scaling.

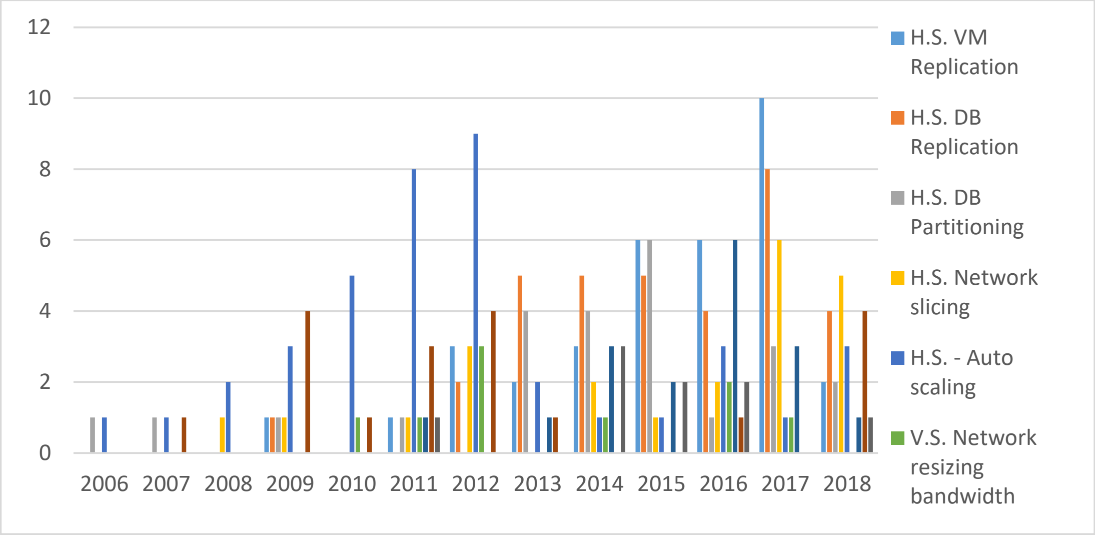


Figure 4.10 Publication numbers for each scalability type

RQ7- What type of research is conducted on cloud scalability?

Table 4.8 and Figure 4.11 show the annual publication number of cloud scalability types while Figure 4.12 shows the percentage of annual publication count of cloud scalability research types. Of the five types under study, we note that Experience papers was the most popular of the types, where it received 71 papers within the period of study, followed by 45 papers for Opinion, 41 papers for Validation, 25 for Philosophical and 16 for Evaluation.

The period between 2016 and 2017 is characterized by the proliferation of cloud propagation papers, as is illustrated in Figure 4.11. The most widespread publication of cloud papers was in 2017, where Validation and Opinion received 8 papers for each and 7 papers for Experience. In 2016, Opinion had 8 papers while Validation and Experience had only 7 papers for each.

Table 4.8 Annual publication count for each research type

Year of publication	Research type				
	Validation	Opinion	Experience	Philosophical	Evaluation
2006		1	1		
2007		1	1	1	
2008	1		3		
2009		2	5	3	
2010	1	1	4	1	
2011	4	3	8	2	
2012	3	4	9	4	3
2013	2	3	4	4	1
2014	4	6	7	1	4
2015	6	6	4	2	3
2016	7	8	7	2	
2017	8	8	7	3	4
2018	5	2	11	2	1
Total	41	45	71	25	16

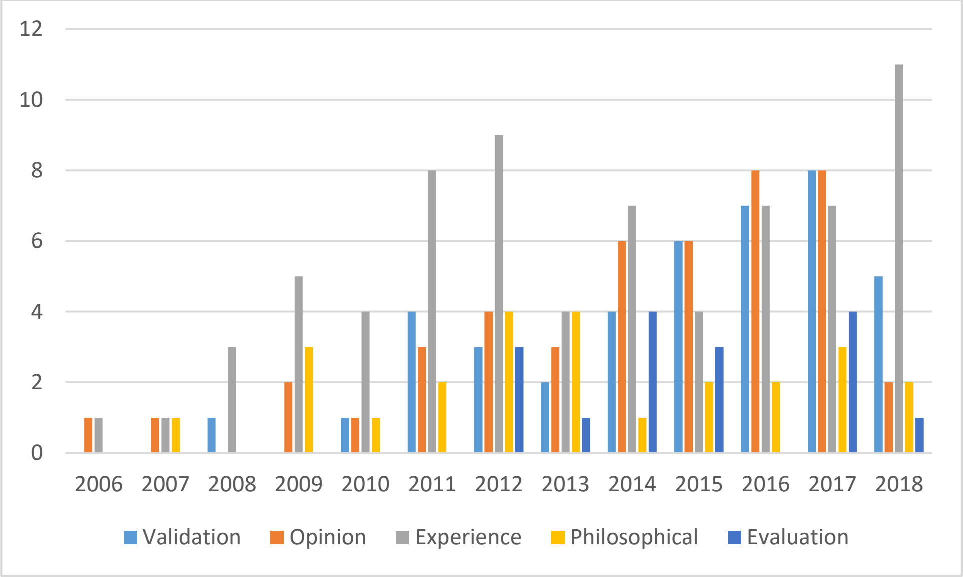


Figure 4.11 Annual publication count for cloud scalability research types

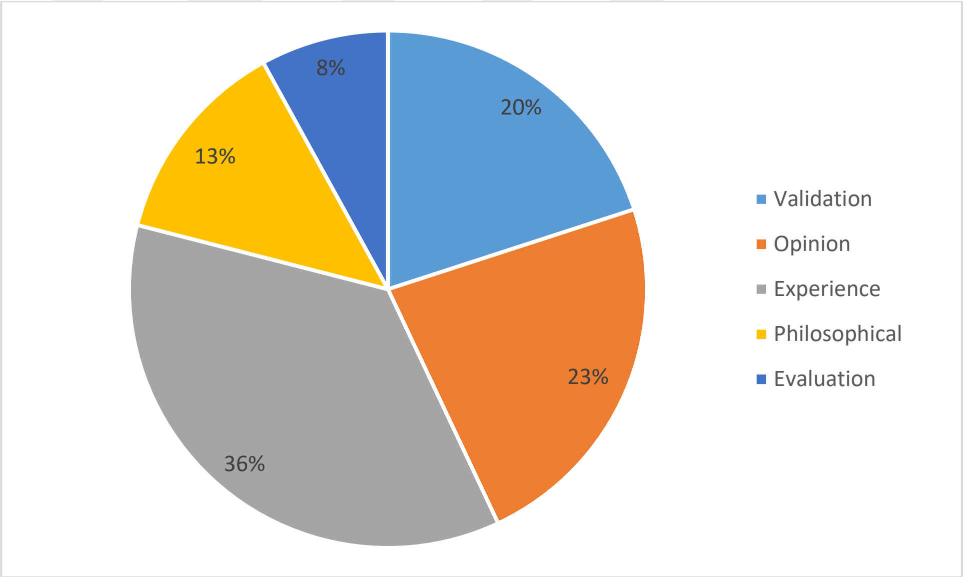


Figure 4.12 Percentage of publication count for cloud scalability research types

CHAPTER 5

CONCLUSION

The objective of this study is to demonstrate the techniques used for scalability in cloud computing for both horizontal and vertical scaling. The systematic mapping study is the method used in this thesis. As a result of the analysis of papers collected from systematic mapping, we can say that horizontal and vertical scalability has different types of methods. In this thesis, 198 papers were reviewed and analyzed to identify answers to the research questions raised in Section 4.1.1 of Chapter 4.

First of foremost, the first two research questions provide information on the databases used in this thesis and information about the venue, whether it is a conference or journal, taking into consideration the number of articles for each year for both horizontal and vertical types. The results of the first question indicated that nearly half of the papers were collected from IEEE Xplore followed by ACM Digital Library at 24% of the total, and the rest were flat between the Science Direct and Google Scholar as shown in Figure 4.2. As for the second question, the results showed that more than half of the papers were for conferences as shown in Figure 4.3. Moreover, the answers indicated that in 2017, contributors made studies of horizontal scalability more than other years, whereas for vertical scalability the year 2016 contained the largest number of studies as shown in Figure 4.4 and Figure 4.5.

The third and fourth questions are about the methods used in both vertical and horizontal types. Those methods that were extracted from the articles published in the period specified in this study were focused on virtual machine / containers, database, network and on auto-scaling. On one hand, resizing methods were used for vertical scalability for either virtual machine / container or database and resizing bandwidth for network scaling. On the other hand, replication methods were used for horizontal scaling for virtual machine / containers. As for the database, two methods were used for horizontal scaling, the first method was replication and the second was partitioning and slicing method were used for horizontal scaling for network.

The results of the research method used in this study show that contributors focused on horizontal scalability much more than vertical scalability. In addition, the concentration in vertical scaling was on the auto scaling at 35.5% and virtual machine at 31.5% out of the vertical scaling. At the same time, the focus of researchers in horizontal scaling on the database was 38% out of horizontal scaling as shown in Figure 4.7 and Figure 4.9.

The last question is about the destination of articles in terms of type of research. Where the answers to this question indicate that most contributors to the scalability of cloud computing are expressing their experience results and the proportion of nearly 36 %. Moreover, around 23% of the contributors in this field expressing their opinion. As for the remaining percentages 41%, there was a 20% for validation, 13% for philosophical and 8% for evaluations as shown in Figure 4.11 and Figure 4.12.

As a result, this master thesis illustrates several trends in the scalability of cloud computing, especially in vertical and horizontal scaling, which can help researchers gain an overview of the field and identify areas that require more attention from the research community.

REFERENCES

[1] N. Anwar, "Architecting Scalable Web Application with Scalable Cloud Platform," 2018.

[2] lintut.com. (2015). master slave. Available: https://lintut.com/tag/master-slave/

[3] BizDevOps. (2019). Horizontal Scaling. Available: http://bizdevops.uk/terminology/concepts/horizontal-scaling/

[4]cloudgirl.tech. (2017). Data Partitioning: Vertical Partitioning, Horizontal Partitioning, and Hybrid Partitioning. Available: http://cloudgirl.tech/data-partitioning-vertical-horizontal-hybrid-partitioning/

[5] Severalnines. (2016). Learn the difference between Multi-Master and Multi-Source replication. Available: https://severalnines.com/blog/learn-difference-between-multi-master-and-multi-source-replication

[6] Diagram. (2019). Saas Paas and Iaas Architecture Diagrams Iaas Vs Paas Vs Saas A Clear Explanation Of Cloud Services. Available: https://diagram.alimb.us/50-saas-paas-and-iaas-architecture-diagrams-nf8p/saas-paas-and-iaas-architecture-diagrams-iaas-vs-paas-vs-saas-a-clear-explanation-of-cloud-services/

[7] C. C. Hosting. (2019). Cloud Load Balancer. Available: https://www.cacloud.com/services/cloud-load-balancer/

[8] S. Ried, H. Kisker, and P. Matzke, "The evolution of cloud computing markets," Forrester Research, 2010.

[9] P. Mell and T. Grance, "The NIST definition of cloud computing," Communications of the ACM, vol. 53, p. 50, 2010.

[10] Amazon Web Services. (2018,). What is Cloud Computing?, . Available: https://aws.amazon.com/tr/what-is-cloud-computing/

[11] Microsoft Azure. (2018,). What is cloud computing?, . Available: https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/

[12] K. S. Mehta, "Analysis of Cloud Computing Security Considerations for Platform as a Service," International Journal of Computer and Communication Engineering, vol. 2, p. 197, 2013.

[13] M. Sajid and Z. Raza, "Cloud computing: Issues & challenges," in International Conference on Cloud, Big Data and Trust, 2013, pp. 13-15.

[14] B. Tang, R. Sandhu, and Q. Li, "Multi-tenancy authorization models for collaborative cloud services," Concurrency and Computation: Practice and Experience, vol. 27, pp. 2851-2868, 2015.

[15] S. Heinzl and C. Metz, "Toward a cloud-ready dynamic load balancer based on the apache web server," in Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2013 IEEE 22nd International Workshop on, 2013, pp. 342-345.

[16] T. Baars, R. Khadka, H. Stefanov, S. Jansen, R. Batenburg, and E. van Heusden, "Chargeback for cloud services," Future Generation Computer Systems, vol. 41, pp. 91-103, 2014.

[17] O. A. Durowoju, H. K. Chan, and X. Wang, "The impact of security and scalability of cloud service on supply chain performance," Journal of Electronic Commerce Research, vol. 12, pp. 243-256, 2011.

[18] D. Moldovan, H.-L. Truong, and S. Dustdar, "Cost-aware scalability of applications in public clouds," in 2016 IEEE International Conference on Cloud Engineering (IC2E), 2016, pp. 79-88.

[19] M. Nazir, "Cloud computing: overview & current research challenges," IOSR journal of computer engineering, vol. 8, pp. 14-22, 2012.

[20] F. Sabahi, "Cloud Computing RAS issues and challenges," ICTer, vol. 4, 2011.

[21] A. Shawish and M. Salama, "Cloud computing: paradigms and technologies," in Inter-cooperative collective intelligence: Techniques and applications, ed: Springer, 2014, pp. 39-67.

[22] S. Namasudra, P. Roy, and B. Balusamy, "Cloud computing: fundamentals and research issues," in Recent Trends and Challenges in Computational Models (ICRTCCM), 2017 Second International Conference on, 2017, pp. 7-12.

[23] A. A.-S. Ahmad and P. Andras, "Measuring the Scalability of Cloud-based Software Services," in 2018 IEEE World Congress on Services (SERVICES), 2018, pp. 5-6.

[24] B. G. Batista, J. C. Estrella, C. H. G. Ferreira, D. M. Leite Filho, L. H. V. Nakamura, S. Reiff-Marganiec, et al., "Performance evaluation of resource management in cloud computing environments," PloS one, vol. 10, p. e0141914, 2015.

[25] D. D. R. Ab Rashid Dar, "Survey On Scalability In Cloud Environment," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 5, 2016.

[26] I. Gangwar and P. Rana, "Cloud Computing Overview: Services and Features," International Journal of Innovations and Advancement in Computer Science, vol. 3, 2014.

[27] L. A. a. G. J. M. M.Kriushanth, "Auto Scaling in Cloud Computing: An Overview," International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, p. 6, 2013.

[28] R. Anandhi and K. Chitra, "A challenge in improving the consistency of transactions in cloud databases-scalability," International Journal of Computer Applications, vol. 52, pp. 12-14, 2012.

[29] wikipedia. (2019,). Autoscaling, . Available: https://en.wikipedia.org/wiki/Autoscaling

[30] B. C. Carroll, "Auto-Scaling in the Cloud: Evaluating a Control Based Technique."

[31] T. Lorido-Botrán, J. Miguel-Alonso, and J. A. Lozano, "Auto-scaling techniques for elastic applications in cloud environments," Department of Computer Architecture and Technology, University of Basque Country, Tech. Rep. EHU-KAT-IK-09, vol. 12, p. 2012, 2012.

[32] J. Dejun, G. Pierre, and C.-H. Chi, "EC2 performance analysis for resource provisioning of service-oriented applications," in Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops, 2009, pp. 197-207.

[33] Y. W. Ahn, A. M. Cheng, J. Baek, M. Jo, and H.-H. Chen, "An auto-scaling mechanism for virtual resources to support mobile, pervasive, real-time healthcare applications in cloud computing," IEEE Network, vol. 27, pp. 62-68, 2013.

[34] M. Mao, J. Li, and M. Humphrey, "Cloud auto-scaling with deadline and budget constraints," in 2010 11th IEEE/ACM International Conference on Grid Computing, 2010, pp. 41-48.

[35] D. C. Erdil, "Autonomic cloud resource sharing for intercloud federations," Future Generation Computer Systems, vol. 29, pp. 1700-1708, 2013.

[36] A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford, "In VINI veritas: realistic and controlled network experimentation," pp. 3-14.

[37] T. Wood, P. J. Shenoy, A. Gerber, J. E. van der Merwe, and K. K. Ramakrishnan, "The Case for Enterprise-Ready Virtual Private Clouds."

[38] L. M. Vaquero, L. Rodero-Merino, and R. Buyya, "Dynamically scaling applications in the cloud," ACM SIGCOMM Computer Communication Review, vol. 41, pp. 45-52, 2011.

[39] I. Baldine, Y. Xin, D. Evans, C. Heerman, J. Chase, V. Marupadi, et al., "The missing link: Putting the network in networked cloud computing."

[40] N. D. Mickulicz, P. Narasimhan, and R. Gandhi, "To auto scale or not to auto scale," in Proceedings of the 10th International Conference on Autonomic Computing ({ICAC} 13), 2013, pp. 145-151.

[41] C. Gokhan, Z. Karakaya, and A. Yazici, "Systematic mapping study on performance scalability in big data on cloud using vm and container," in IFIP International Conference on Artificial Intelligence Applications and Innovations, 2016, pp. 634-641.

[42] G. Ogutcu, A Systematic Mapping Study on Technology Transfer vol. 3, 2017.

[43] S. Q. Fleh, O. Bayat, S. Al-Azawi, and O. N. Ucan, "A systematic mapping study on touch classification," International Journal of Computer Science and Network Security, vol. 18, p. 7, 2018.

[44] Y. Tjong, S. Adi, R. Kosala, and H. Prabowo, A systematic mapping study on enterprise architecture framework for HEI vol. 9, 2018.

[45] J. L. Barros-Justo, A. L. Cravero-Leal, F. B. Benitti, and R. Capilla-Sevilla, "Systematic Mapping Protocol: The impact of using software patterns during requirements engineering activities in real-world settings," arXiv preprint arXiv:1701.05747, 2017.

[46] T. Kosar, S. Bohra, and M. Mernik, "Protocol of a systematic mapping study for domain-specific languages," Journal of Information and Software Technology, vol. 21, pp. 77-91, 2016.

[47] V. Dos Santos, É. F. De Souza, K. R. Felizardo, and N. L. Vijaykumar, "Analyzing the Use of Concept Maps in Computer Science: A Systematic Mapping Study," Informatics in Education, vol. 16, pp. 257-288, 2017.

[48] R. d. A. Mauricio, L. Veado, R. T. Moreira, E. Figueiredo, and H. Costa, "A systematic mapping study on game-related methods for software engineering education," Information and software technology, vol. 95, pp. 201-218, 2018.

[49] I. Odun-Ayo, R. Goddy-Worlu, O. Ajayi, and E. Grant, "A Systematic Mapping Study of High Performance Computing and the Cloud," ARPN Journal of Engineering and Applied Sciences, vol. 13, pp. 9886-9700, 2018.

[50] I. Odun-Ayo, O. Ajayi, R. Goddy-Worlu, and J. Yahaya, "A Systematic Mapping Study of Cloud Resources Management and Scalability in Brokering, Scheduling, Capacity Planning and Elasticity," Asian Journal of Scientific Research, 2019.

[51] A. Yazici, Z. Karakaya, and M. Alayyoub, "A Systematic Study for Big Data Stream Processing Frameworks," Journal on Advances in Theoretical and Applied Informatics, vol. 2, pp. 4-11, 2016.

[52] C.-Y. Liu, M.-R. Shie, Y.-F. Lee, Y.-C. Lin, and K.-C. Lai, "Vertical/horizontal resource scaling mechanism for federated clouds," in 2014 International Conference on Information Science & Applications (ICISA), 2014, pp. 1-4.

[53] S. Sotiriadis, N. Bessis, C. Amza, and R. Buyya, "Vertical and horizontal elasticity for dynamic virtual machine reconfiguration," IEEE Transactions on Services Computing, pp. 1-1, 2016.

[54] K. Hwang, X. Bai, Y. Shi, M. Li, W.-G. Chen, and Y. Wu, "Cloud performance modeling with benchmark evaluation of elastic scaling strategies," IEEE Transactions on Parallel and Distributed Systems, vol. 27, pp. 130-143, 2016.

[55] H. Alipour, Y. Liu, and A. Hamou-Lhadj, "Analyzing auto-scaling issues in cloud environments," in Proceedings of 24th Annual International Conference on Computer Science and Software Engineering, 2014, pp. 75-89.

[56] M. M. Falatah and O. A. Batarfi, "Cloud scalability considerations," International Journal of Computer Science and Engineering Survey, vol. 5, p. 37, 2014.

[57] C.-L. Hung, Y.-C. Hu, and K.-C. Li, "Auto-scaling model for cloud computing system," International Journal of Hybrid Information Technology, vol. 5, pp. 181-186, 2012.

[58] T. C. Chieu, A. Mohindra, A. A. Karve, and A. Segal, "Dynamic scaling of web applications in a virtualized cloud computing environment," in 2009 IEEE International Conference on e-Business Engineering, 2009, pp. 281-286.

[59] S. V. Nandgaonkar and A. Raut, "A comprehensive study on cloud computing," International Journal of Computer Science and Mobile Computing, vol. 3, pp. 733-738, 2014.

[60] W. Iqbal, M. N. Dailey, D. Carrera, and P. Janecek, "Adaptive resource provisioning for read intensive multi-tier applications in the cloud," Future Generation Computer Systems, vol. 27, pp. 871-879, 2011.

[61] H. Vural, M. Koyuncu, and S. Guney, "A systematic literature review on microservices," in International Conference on Computational Science and Its Applications, 2017, pp. 203-217.

[62] I. Odun-Ayo, R. Goddy-Worlu, V. Geteloma, and E. Grant, "A Systematic Mapping Study of Cloud, Fog, and Edge/Mobile Devices Management, Hierarchy Models and Business Models," Advances in Science, Technology and Engineering Systems Journal, vol. 4, pp. 91-101, 2018.

[63] A. Almishal and A. Youssef, Cloud Service Providers: A Comparative Study vol. 5, 2014.

[64] A. W. servives. (2019). AWS. Available: aws.amazon.com

[65] docs.aws.amazon.com. (2019). Partitioning Data. Available: https://docs.aws.amazon.com/athena/latest/ug/partitions.html

[66] A. W. Services. (2019). Ongoing Data Replicatio. Available: https://d1.awsstatic.com/Marketplace/scenarios/bi/Q42017/BIA21-ongoing-data-replication-Scenario-Brief.pdf

[67] A. Bala and I. Chana, "Fault tolerance-challenges, techniques and implementation in cloud computing," International Journal of Computer Science Issues (IJCSI), vol. 9, p. 288, 2012.

[68] W. Azure. (2019). Available: www.windowsazure.com

[69] S. Agrawal, S. Chaudhuri, L. Kollar, A. Marathe, V. Narasayya, and M. Syamala, "Database tuning advisor for microsoft sql server 2005," pp. 930-932.

[70] M. Rys, "scalable SQL," Communications of the ACM, vol. 54, pp. 48-53, 2011.

[71] A. Bartolo. (2013). Step-By-Step: Virtual Machine Replication Using Hyper-V Replica. Available: https://blogs.technet.microsoft.com/canitpro/2013/04/07/step-by-step-virtual-machine-replication-using-hyper-v-replica/

[72] Drew McDaniel. (2016,). Resize virtual machines, . Available: https://azure.microsoft.com/tr-tr/blog/resize-virtual-machines/

[73] Google cloud. (2019,). Build. Modernize. Scale. , . Available: https://cloud.google.com

[74] Google Cloud. (2019,). Replication options , . Available: https://cloud.google.com/sql/docs/mysql/replication/

[75] Google cloud. (2019,). Creating read replicas. Available: https://cloud.google.com/sql/docs/mysql/replication/create-replica

[76] Google cloud. (2019,). Introduction to partitioned tables, . Available: https://cloud.google.com/bigquery/docs/partitioned-tables

[77] Google cloud. (2019,). Applying Sizing Recommendations for VM Instances, . Available:https://cloud.google.com/compute/docs/instances/apply-sizing-recommendations-for-instances

APPENDICES

Appendix A: Collected Data for Cloud Scalability

paper type						scalability type								
						horizontal					vertical			
						Network	virtual machine or container	Auto scaling	database		Network	virtual machine or container	Auto scaling	database
reference	database	journal	conference	year	research type	slicing	replication	?????	replication	partitioning	resizing bandwidth	resizing	?????	resizing
A1.	GS	*		2006	Opinion					*				
A2.	IEEE		*	2006	Experience			*						
A3.	GS	*		2007	Opinion					*				
A4.	IEEE		*	2007	Philosophical								*	
A5.	IEEE		*	2007	Experience			*						
A6.	GS	*		2008	Experience	*								
A7.	ACM	*		2008	Validation			*						
A8.	GS		*	2008	Experience			*						
A9.	GS	*		2009	Experience			*						
A10.	ACM	*		2009	Experience				*					
A11.	ACM		*	2009	Philosophical								*	
A12.	IEEE		*	2009	Experience								*	
A13.	GS	*		2009	Experience			*						
A14.	ACM		*	2009	Philosophical								*	
A15.	IEEE		*	2009	Opinion		*							
A16.	ACM		*	2009	Experience			*						
A17.	SD	*		2009	Philosophical					*				
A18.	ACM		*	2009	Opinion								*	
A19.	IEEE		*	2010	Experience			*						
A20.	IEEE		*	2010	Philosophical						*			
A21.	ACM		*	2010	Validation			*						

A22.	IEEE		*	2010	Experience			*						
A23.	IEEE		*	2010	Opinion								*	
A24.	IEEE		*	2010	Experience			*						
A25.	IEEE		*	2010	Experience			*						
A26.	GS	*		2011	Validation						*			
A27.	IEEE		*	2011	Experience			*						
A28.	IEEE		*	2011	Validation		*							
A29.	ACM		*	2011	Experience								*	
A30.	ACM		*	2011	Validation			*						
A31.	IEEE		*	2011	Experience								*	
A32.	IEEE		*	2011	Philosophical			*						
A33.	ACM	*		2011	Experience			*						
A34.	ACM	*		2011	Opinion					*				
A35.	GS		*	2011	Experience								*	
A36.	IEEE		*	2011	Opinion			*						
A37.	ACM		*	2011	Experience	*								
A38.	IEEE		*	2011	Experience			*						
A39.	ACM		*	2011	Validation									*
A40.	GS		*	2011	Philosophical			*						
A41.	ACM		*	2011	Opinion							*		
A42.	GS	*		2011	Experience			*						
A43.	GS	*		2012	Validation			*						
A44.	IEEE	*		2012	Opinion				*					
A45.	IEEE		*	2012	Experience						*			
A46.	ACM		*	2012	Philosophical			*						
A47.	IEEE		*	2012	Opinion			*						
A48.	IEEE		*	2012	Experience			*						
A49.	ACM		*	2012	Experience	*								
A50.	ACM		*	2012	Experience			*						
A51.	ACM		*	2012	Experience						*			
A52.	IEEE		*	2012	Opinion		*							
A53.	GS	*		2012	Experience								*	
A45.	IEEE		*	2012	Philosophical						*			
A54.	IEEE		*	2012	Experience								*	
A55.	IEEE		*	2012	Experience			*						

A56.	IEEE	*		2012	Evaluation								*	
A57.	ACM	*		2012	Evaluation		*							
A58.	IEEE		*	2012	Evaluation			*						
A59.	ACM		*	2012	Experience			*						
A60.	IEEE	*		2012	Validation								*	
A61.	GS		*	2012	Experience	*								
A62.	IEEE		*	2012	Validation		*							
A63.	SD	*		2012	Opinion	*								
A64.	IEEE		*	2012	Philosophical				*					
A65.	GS		*	2012	Philosophical			*						
A66.	ACM	*		2013	Experience								*	
A67.	ACM		*	2013	Validation				*					
A68.	SD	*		2013	Evaluation				*	*				
A69.	GS	*		2013	Experience			*						
A70.	IEEE		*	2013	Opinion				*					
A71.	IEEE		*	2013	Philosophical					*				
A72.	ACM		*	2013	Experience			*						
A73.	IEEE		*	2013	Experience					*				
A74.	IEEE	*		2013	Validation				*					
A75.	GS		*	2013	Philosophical				*					
A76.	SD	*		2013	Philosophical		*							
A77.	SD	*		2013	Philosophical					*				
A78.	IEEE		*	2013	Opinion							*		
A79.	ACM		*	2013	Opinion		*							
A80.	IEEE		*	2014	Experience							*		
A81.	IEEE		*	2014	Evaluation		*							
A82.	IEEE		*	2014	Evaluation					*				
A83.	GS	*		2014	Experience			*						
A84.	GS		*	2014	Validation									*
A85.	IEEE		*	2014	Evaluation		*							
A86.	SD	*		2014	Experience				*					
A87.	SD	*		2014	Experience	*								
A88.	ACM		*	2014	Evaluation				*					
A89.	S.D	*		2014	Experience				*					
A90.	SD	*		2014	Opinion							*		
A91.	ACM		*	2014	Experience						*			
A92.	SD	*		2014	Opinion		*							
A93.	IEEE		*	2014	Validation				*					
A90.	SD	*		2014	Opinion									*

A94.	IEEE		*	2014	Opinion					*				
A95.	ACM		*	2014	Validation							*		
A96.	IEEE		*	2014	Philosophical					*				
A97.	ACM		*	2014	Experience	*								
A98.	IEEE		*	2014	Opinion									*
A99.	ACM		*	2014	Opinion					*				
A100.	IEEE		*	2014	Validation				*					
A101.	IEEE	*		2015	Experience				*					
A102.	IEEE		*	2015	Validation				*					
A103.	IEEE		*	2015	Opinion				*					
A104.	SD	*		2015	Evaluation				*					
A105.	SD	*		2015	Experience			*						
A106.	IEEE	*		2015	Validation				*					
A107.	SD	*		2015	Opinion					*				
A108.	IEEE		*	2015	Philosophical		*							
A109.	ACM		*	2015	Evaluation					*				
A110.	SD	*		2015	Evaluation							*		
A111.	ACM		*	2015	Philosophical					*				
A112.	ACM		*	2015	Experience	*								
A113.	GS	*		2015	Validation		*							
A114.	IEEE	*		2015	Opinion		*							
A115.	ACM	*		2015	Opinion					*				
A116.	SD	*		2015	Validation									*
A117.	ACM	*		2015	Philosophical		*							
A118.	IEEE		*	2015	Validation					*				
A119.	ACM		*	2015	Validation		*							
A120.	SD	*		2015	Philosophical									*
A121.	GS		*	2015	Opinion					*				
A122.	S.D	*		2015	Opinion		*							
A123.	ACM		*	2015	Experience							*		
A124.	GS	*		2016	Validation		*							
A125.	IEEE		*	2016	Experience	*								
A126.	ACM		*	2016	Validation		*							
A127.	SD	*		2016	Experience			*						
A128.	GS		*	2016	Validation					*				
A129.	SD		*	2016	Validation									*
A130.	SD	*		2016	Validation								*	
A131.	SD	*		2016	Philosophical		*							
A132.	ACM		*	2016	Experience	*								
A130.	SD	*		2016	Philosophical							*		

A133.	IEEE		*	2016	Opinion				*					
A134.	GS		*	2016	Opinion									*
A135.	IEEE		*	2016	Experience			*						
A136.	IEEE		*	2016	Opinion				*					
A137.	IEEE		*	2016	Opinion							*		
A138.	GS	*		2016	Experience						*			
A139.	IEEE		*	2016	Experience						*			
A140.	SD	*		2016	Philosophical							*		
A141.	SD	*		2016	Philosophical				*					
A142.	IEEE		*	2016	Opinion							*		
A143.	SD	*		2016	Experience			*						
A144.	ACM	*		2016	Validation							*		
A145.	ACM		*	2016	Opinion		*							
A146.	IEEE	*		2016	Opinion		*					*		
A147.	SD	*		2016	Validation				*					
A148.	IEEE		*	2017	Validation		*							
A149.	ACM	*		2017	Evaluation		*							
A150.	IEEE		*	2017	Opinion					*				
A151.	SD	*		2017	Experience			*						
A152.	SD	*		2017	Validation				*					
A153.	SD	*		2017	Evaluation				*					
A154.	ACM	*		2017	Validation		*							
A155.	SD	*		2017	Opinion		*					*		
A156.	IEEE		*	2017	Validation		*							
A157.	GS		*	2017	Opinion				*					
A158.	IEEE	*		2017	Philosophical		*							
A159.	SD	*		2017	Experience				*					
A160.	SD	*		2017	Validation				*					
A161.	IEEE	*		2017	Experience	*								
A162.	IEEE	*		2017	Experience	*								
A163.	IEEE		*	2017	Experience	*								
A164.	IEEE		*	2017	Philosophical	*								
A165.	IEEE		*	2017	Opinion				*					
A166.	SD	*		2017	Philosophical				*					
A167.	ACM		*	2017	Validation	*								
A168.	IEEE		*	2017	Opinion					*				
A169.	IEEE		*	2017	Opinion				*					
A170.	ACM		*	2017	Validation		*							

A171.	SD	*		2017	Opinion	*								
A172.	IEEE		*	2017	Opinion		*							
A173.	IEEE		*	2017	Evaluation							*		
A174.	IEEE	*		2017	Validation							*		
A175.	IEEE		*	2017	Experience						*			
A176.	SD	*		2017	Experience		*							
A177.	IEEE	*		2017	Evaluation					*				
A178.	S.D	*		2018	Experience					*				
A179.	IEEE		*	2018	Experience	*								
A180.	SD	*		2018	Validation								*	
A181.	ACM	*		2018	Validation		*							
A182.	IEEE		*	2018	Experience	*								
A183.	IEEE		*	2018	Opinion		*							
A184.	SD	*		2018	Experience								*	
A185.	SD	*		2018	Experience								*	
A186.	SD	*		2018	Experience			*						
A187.	SD	*		2018	Validation			*						
A188.	SD	*		2018	Experience								*	
A189.	IEEE		*	2018	Philosophical				*					
A190.	IEEE	*		2018	Philosophical				*	*				
A191.	IEEE	*		2018	Experience	*								
A192.	SD	*		2018	Validation				*					
A193.	GS	*		2018	Evaluation							*		
A194.	SD	*		2018	Opinion	*								
A195.	SD	*		2018	Validation				*					
A196.	SD	*		2018	Experience			*						
A197.	ACM		*	2018	Experience					*				*
A198.	ACM		*	2018	Experience	*								

Appendix B: List of Papers

A 1. Watson, M.C. and P.Y. Colin, *Method and apparatus for partitioning data for storage in a database*. 2006, Google Patents.

A 2. Tesauro, G., et al. *A hybrid reinforcement learning approach to autonomic resource allocation*. IEEE.

A 3. Dumler, M., *Microsoft SQL Server 2008 Product Overview*. 2007.

A 4. Xu, J., et al. *On the use of fuzzy modeling in virtualized data center management*. IEEE.

A 5. Zhang, Q., L. Cherkasova, and E. Smirni. *A regression-based analytic model for dynamic resource provisioning of multi-tier applications*. IEEE.

A 6. Alice, A.L., P.M.C. Nisha, and S. Sivagami, *Enhancing Security of Multi-cloud Architecture using combination of approaches.* 2015.

A 7. Urgaonkar, B., et al., *Agile dynamic provisioning of multi-tier internet applications.* ACM Transactions on Autonomous and Adaptive Systems (TAAS), 2008. **3**(1): p. 1.

A 8. Chen, G., et al. *Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services*.

A 9. Bodík, P., et al., *Statistical Machine Learning Makes Automatic Control Practical for Internet Datacenters.* HotCloud, 2009. **9**: p. 12-12.

A 10. Gueye, M., I. Sarr, and S. Ndiaye, *Database replication in large scale systems: optimizing the number of replicas*, in *Proceedings of the 2009 EDBT/ICDT Workshops*. 2009, ACM: Saint-Petersburg, Russia. p. 3-9.

A 11. Padala, P., et al. *Automated control of multiple virtualized resources*. ACM.

A 12. Park, S.-M. and M. Humphrey. *Self-tuning virtual machines for predictable escience*. IEEE.

A 13. Prodan, R. and V. Nae, *Prediction-based real-time resource provisioning for massively multiplayer online games.* Future Generation Computer Systems, 2009. **25**(7): p. 785-793.

A 14. Rao, J., et al. *VCONF: a reinforcement learning approach to virtual machines auto-configuration*. ACM.

A 15. Du, Y. and H. Yu. *Paratus: Instantaneous failover via virtual machine replication*. in *2009 Eighth International Conference on Grid and Cooperative Computing*. 2009. IEEE.

A 16. Lim, H.C., et al. *Automated control in cloud computing: challenges and opportunities*. ACM.

A 17. He, Z. and P. Veeraraghavan, *Fine-grained updates in database management systems for flash memory*. Information Sciences, 2009. **179**(18): p. 3162-3181.

A 18. Kalyvianaki, E., T. Charalambous, and S. Hand. *Self-adaptive and self-configured CPU resource provisioning for virtualized servers using Kalman filters*. ACM.

A 19. Khatua, S., A. Ghosh, and N. Mukherjee. *Optimizing the utilization of virtual resources in cloud environment*. IEEE.

A 20. Beloglazov, A. and R. Buyya. *Energy efficient resource management in virtualized cloud data centers*. IEEE Computer Society.

A 21. Lim, H.C., S. Babu, and J.S. Chase. *Automated control for elastic storage*. ACM.

A 22. Bacigalupo, D.A., et al. *Resource management of enterprise cloud systems using layered queuing and historical performance models*. IEEE.

A 23. Gong, Z., X. Gu, and J. Wilkes. *Press: Predictive elastic resource scaling for cloud systems*. Ieee.

A 24. Mi, H., et al. *Online Self-Reconfiguration with Performance Guarantee for Energy-Efficient Large-Scale Cloud Computing Data Centers*. IEEE Computer Society.

A 25. Dutreilh, X., et al. *From data center resource allocation to control theory and back*. IEEE.

A 26. Rodrigues, H., et al., *Gatekeeper: Supporting Bandwidth Guarantees for Multi-tenant Datacenter Networks.* WIOV, 2011. **1**(3): p. 784-789.

A 27. Roy, N., A. Dubey, and A. Gokhale. *Efficient autoscaling in the cloud using predictive models for workload forecasting*. IEEE.

A 28. Gerofi, B. and Y. Ishikawa. *Rdma based replication of multiprocessor virtual machines over high-performance interconnects*. in *2011 IEEE International Conference on Cluster Computing*. 2011. IEEE.

A 29. Shen, Z., et al. *Cloudscale: elastic resource scaling for multi-tenant cloud systems*. ACM.

A 30. Simmons, B., et al. *Managing a SaaS application in the cloud using PaaS policy sets and a strategy-tree*. International Federation for Information Processing.

A 31. Wang, L., et al. *Fuzzy modeling based resource management for virtualized database systems*. IEEE.

A 32. Rao, J., et al. *A distributed self-learning approach for elastic provisioning of virtualized cloud resources*. IEEE.

A 33. Caron, E., F. Desprez, and A. Muresan, *Pattern Matching Based Forecast of Non-periodic Repetitive Behavior for Cloud Clients.* Journal of Grid Computing, 2011. **9**(1): p. 49-64.

A 34. Mansour, E., et al., *ERA: efficient serial and parallel suffix tree construction for very long strings.* Proc. VLDB Endow., 2011. **5**(1): p. 49-60.

A 35. Maurer, M., I. Brandic, and R. Sakellariou. *Enacting SLAs in clouds using rules*. Springer.

A 36. Ghanbari, H., et al. *Exploring alternative approaches to implement an elasticity policy*. IEEE.

A 37. Gamage, S., et al. *Opportunistic flooding to improve TCP transmit performance in virtualized clouds*. ACM.

A 38. Chieu, T.C., A. Mohindra, and A.A. Karve. *Scalability and performance of web applications in a compute cloud*. IEEE.

A 39. Pokorny, J., *NoSQL databases: a step to database scalability in web environment*, in *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services*. 2011, ACM: Ho Chi Minh City, Vietnam. p. 278-283.

A 40. Dutreilh, X., et al. *Using reinforcement learning for autonomic resource allocation in clouds: towards a fully automated workflow*.

A 41. Al-Kiswany, S., et al., *VMFlock: virtual machine co-migration for the cloud*, in *Proceedings of the 20th international symposium on High performance distributed computing*. 2011, ACM: San Jose, California, USA. p. 159-170.

A 42. Iqbal, W., et al., *Adaptive resource provisioning for read intensive multi-tier applications in the cloud.* Future Generation Computer Systems, 2011. **27**(6): p. 871-879.

A 43. Islam, S., et al., *Empirical prediction models for adaptive resource provisioning in the cloud.* Future Generation Computer Systems, 2012. **28**(1): p. 155-162.

A 44. Arrieta-Salinas, I., J.E. Armend'riz-Iñigo, and J. Navarro. *Classic replication techniques on the cloud*. in *2012 Seventh International Conference on Availability, Reliability and Security*. 2012. IEEE.

A 45. Feller, E., L. Rilling, and C. Morin. *Snooze: A scalable and autonomic virtual machine management framework for private clouds*. IEEE.

A 46. Ali-Eldin, A., et al. *Efficient provisioning of bursty scientific workloads on the cloud using adaptive elasticity control*. ACM.

A 47. Ali-Eldin, A., J. Tordsson, and E. Elmroth. *An adaptive hybrid elasticity controller for cloud infrastructures*. IEEE.

A 48. Dutta, S., et al. *Smartscale: Automatic application scaling in enterprise clouds*. IEEE.

A 49. Heller, B., R. Sherwood, and N. McKeown. *The controller placement problem*. ACM.

A 50. Han, R., et al. *Lightweight resource scaling for cloud applications*. IEEE Computer Society.

A 51. Hillenbrand, M., et al. *Virtual InfiniBand clusters for HPC clouds*. ACM.

A 52. Wang, W., H. Chen, and X. Chen. *An availability-aware virtual machine placement approach for dynamic scaling of cloud applications*. in *2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing*. 2012. IEEE.

A 53. Xu, C.-Z., J. Rao, and X. Bu, *URL: A unified reinforcement learning approach for autonomic cloud management.* Journal of Parallel and Distributed Computing, 2012. **72**(2): p. 95-105.

A 54. Hasan, M.Z., et al. *Integrated and autonomic cloud resource scaling*. IEEE.

A 55. Huang, J., C. Li, and J. Yu. *Resource prediction based on double exponential smoothing in cloud computing*. IEEE.

A 56. Zhu, Q. and G. Agrawal, *Resource provisioning with budget constraints for adaptive applications in cloud environments.* IEEE Transactions on Services Computing, 2012. **5**(4): p. 497-511.

A 57. Araujo, F., et al., *Replication for dependability on virtualized cloud environments*, in *Proceedings of the 10th International Workshop on Middleware for Grids, Clouds and e-Science*. 2012, ACM: Montreal, Quebec, Canada. p. 1-6.

A 58. Fang, W., et al. *Rpps: A novel resource prediction and provisioning scheme in cloud data center*. IEEE.

A 59. Gambi, A. and G. Toffetti. *Modeling cloud performance with kriging*. IEEE Press.

A 60. Bu, X., J. Rao, and C.-Z. Xu, *Coordinated self-configuration of virtual machines and appliances using a model-free learning approach.* IEEE transactions on parallel and distributed systems, 2012. **24**(4): p. 681-690.

A 61. Naudts, B., et al. *Techno-economic analysis of software defined networking as architecture for the virtualiazation of a mobile network.*

A 62. Wang, W., H. Chen, and X. Chen. *An availability-aware approach to resource placement of dynamic scaling in clouds*. in *2012 IEEE Fifth International Conference on Cloud Computing*. 2012. IEEE.

A 63. Lu, C., et al., *Multi-scale modeling of shock interaction with a cloud of particles using an artificial neural network for model representation.* Procedia IUTAM, 2012. **3**: p. 25-52.

A 64. Zhao, L., S. Sakr, and A. Liu. *Application-managed replication controller for cloud-hosted databases*. in *2012 IEEE Fifth International Conference on Cloud Computing*. 2012. IEEE.

A 65. Koperek, P. and W. Funika. *Dynamic business metrics-driven resource provisioning in cloud environments*. Springer.

A 66. Lama, P. and X. Zhou, *Autonomic provisioning with self-adaptive neural fuzzy control for percentile-based delay guarantee.* ACM Transactions on Autonomous and Adaptive Systems (TAAS), 2013. **8**(2): p. 9.

A 67. R, U., et al., *Robust snapshot replication*, in *Proceedings of the Twenty-Fourth Australasian Database Conference - Volume 137*. 2013, Australian Computer Society, Inc.: Adelaide, Australia. p. 81-91.

A 68. Alam, B., et al., *5-Layered Architecture of Cloud Database Management System.* AASRI Procedia, 2013. **5**: p. 194-199.

A 69. Barrett, E., E. Howley, and J. Duggan, *Applying reinforcement learning towards automating resource allocation and application scalability in the cloud.* Concurrency and Computation: Practice and Experience, 2013. **25**(12): p. 1656-1674.

A 70. Boru, D., et al. *Energy-efficient data replication in cloud computing datacenters*. in *2013 IEEE Globecom Workshops (GC Wkshps)*. 2013. IEEE.

A 71. Ren, P., W. Liu, and D. Sun. *Partition-based data cube storage and parallel queries for cloud computing*. in *2013 Ninth International Conference on Natural Computation (ICNC)*. 2013.

A 72. Chandra, A., W. Gong, and P. Shenoy. *Dynamic resource allocation for shared data centers using online measurements*. Springer-Verlag.

A 73. Lee, K., et al. *Efficient and Customizable Data Partitioning Framework for Distributed Big RDF Data Processing in the Cloud*. in *2013 IEEE Sixth International Conference on Cloud Computing*. 2013.

A 74. Lin, J.-W., C.-H. Chen, and J.M. Chang, *QoS-aware data replication for data-intensive applications in cloud computing systems.* IEEE Transactions on Cloud Computing, 2013. **1**(1): p. 101-115.

A 75. Ardekani, M.S., et al. *On the scalability of snapshot isolation*. in *European Conference on Parallel Processing*. 2013. Springer.

A 76. Gerofi, B., Z. Vass, and Y. Ishikawa, *Utilizing memory content similarity for improving the performance of highly available virtual machines.* Future Generation Computer Systems, 2013. **29**(4): p. 1085-1095.

A 77. Tseng, F.-C., *Mining frequent itemsets in large databases: The hierarchical partitioning approach.* Expert Systems with Applications, 2013. **40**(5): p. 1654-1661.

A 78. Bratterud, A. and H. Haugerud. *Maximizing hypervisor scalability using minimal virtual machines*. in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*. 2013. IEEE.

A 79. Dong, Y., et al., *COLO: COarse-grained LOck-stepping virtual machines for non-stop service*, in *Proceedings of the 4th annual Symposium on Cloud Computing*. 2013, ACM: Santa Clara, California. p. 1-16.

A 80. Lu, L., et al. *Application-driven dynamic vertical scaling of virtual machines in resource pools*. in *2014 IEEE Network Operations and Management Symposium (NOMS)*. 2014. IEEE.

A 81. Glatard, T., et al. *Controlling the deployment of virtual machines on clusters and clouds for scientific computing in CBRAIN*. in *2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. 2014. IEEE.

A 82. Kohler, J. and T. Specht. *Vertical query-join benchmark in a cloud database environment*. in *2014 Second World Conference on Complex Systems (WCCS)*. 2014.

A 83. Han, R., et al., *Enabling cost-aware and adaptive elasticity of multi-tier cloud applications.* Future Generation Computer Systems, 2014. **32**: p. 82-98.

A 84. Mohamed, M.A., O.G. Altrafi, and M.O. Ismail, *Relational vs. nosql databases: A survey.* International Journal of Computer and Information Technology, 2014. **3**(03): p. 598-601.

A 85. Mondal, S.K., et al. *Computing defects per million in cloud caused by virtual machine failures with replication*. in *2014 IEEE 20th Pacific Rim International Symposium on Dependable Computing*. 2014. IEEE.

A 86. Frühwirt, P., et al., *Towards a forensic-aware database solution: Using a secured database replication protocol and transaction management for digital investigations.* Digital Investigation, 2014. **11**(4): p. 336-348.

A 87. Hammadi, A. and L. Mhamdi, *A survey on architectures and energy efficiency in Data Center Networks.* Computer Communications, 2014. **40**: p. 1-21.

A 88. Dhamane, R., et al., *Performance evaluation of database replication systems*, in *Proceedings of the 18th International Database Engineering &#38; Applications Symposium*. 2014, ACM: Porto, Portugal. p. 288-293.

A 89. Zeng, Z. and B. Veeravalli, *Optimal metadata replications and request balancing strategy on cloud data centers.* Journal of Parallel and Distributed Computing, 2014. **74**(10): p. 2934-2940.

A 90. Tesfatsion, S.K., E. Wadbro, and J. Tordsson, *A combined frequency scaling and application elasticity approach for energy-efficient cloud computing.* Sustainable Computing: Informatics and Systems, 2014. **4**(4): p. 205-214.

A 91. Lee, J., et al. *Application-driven bandwidth guarantees in datacenters*. ACM.

A 92. Zeng, L. and Y. Wang, *Optimization on content service with local search in cloud of clouds.* Journal of Network and Computer Applications, 2014. **40**: p. 206-215.

A 93. He, Y., et al. *Scalability analysis and improvement of hadoop virtual cluster with cost consideration*. in *2014 IEEE 7th International Conference on Cloud Computing*. 2014. IEEE.

A 94. Kamal, J.M.M., M. Murshed, and M.M. Gaber. *Predicting Hot-Spots in Distributed Cloud Databases Using Association Rule Mining*. in *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*. 2014.

A 95. Verma, A., J. Bagrodia, and V. Jaiswal. *Virtual machine consolidation in the wild*. in *Proceedings of the 15th International Middleware Conference*. 2014. ACM.

A 96. Li, L. and L. Gruenwald. *SMOPD-C: An autonomous vertical partitioning technique for distributed databases on cluster computers*. in *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*. 2014. IEEE.

A 97. Al-Shabibi, A., et al. *OpenVirteX: Make your virtual SDNs programmable*. ACM.

A 98. Liu, C.-Y., et al. *Vertical/horizontal resource scaling mechanism for federated clouds*. in *2014 International Conference on Information Science & Applications (ICISA)*. 2014. IEEE.

A 99. Aluç, G., M.T. Özsu, and K. Daudjee, *Workload matters: Why RDF databases need a new design.* Proceedings of the VLDB Endowment, 2014. **7**(10): p. 837-840.

A 100. Spaho, E., L. Barolli, and F. Xhafa. *Data replication strategies in P2P systems: A survey*. in *2014 17th International Conference on Network-Based Information Systems*. 2014. IEEE.

A 101. Zhao, L., S. Sakr, and A. Liu, *A framework for consumer-centric SLA management of cloud-hosted databases.* IEEE Transactions on Services Computing, 2015. **8**(4): p. 534-549.

A 102. Garg, A. and S. Bagga. *An autonomic approach for fault tolerance using scaling, replication and monitoring in cloud computing*. in *2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE)*. 2015. IEEE.

A 103. Boru, D., et al. *Models for efficient data replication in cloud computing datacenters*. in *2015 IEEE International Conference on Communications (ICC)*. 2015. IEEE.

A 104. Di Sanzo, P., et al., *A flexible framework for accurate simulation of cloud in-memory data stores.* Simulation Modelling Practice and Theory, 2015. **58**: p. 219-238.

A 105. Hussin, M., N. Asilah Wati Abdul Hamid, and K.A. Kasmiran, *Improving reliability in resource management through adaptive reinforcement learning for distributed systems.* Journal of Parallel and Distributed Computing, 2015. **75**: p. 93-100.

A 106. Barsoum, A.F. and M.A. Hasan, *Provable multicopy dynamic data possession in cloud computing systems.* IEEE Transactions on Information Forensics and Security, 2015. **10**(3): p. 485-497.

A 107. Ogunde, A.O., O. Folorunso, and A.S. Sodiya, *A partition enhanced mining algorithm for distributed association rule mining systems.* Egyptian Informatics Journal, 2015. **16**(3): p. 297-307.

A 108. Zhao, M., et al. *Multi-level VM replication based survivability for mission-critical cloud computing*. in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 2015. IEEE.

A 109. Zhao, W., Y. Cheng, and F. Rusu. *Vertical partitioning for query processing over raw data*. in *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*. 2015. ACM.

A 110. Raghunath, B.R. and B. Annappa, *Virtual Machine Migration Triggering using Application Workload Prediction.* Procedia Computer Science, 2015. **54**: p. 167-176.

A 111. Abelló, A. *Big data design*. in *Proceedings of the ACM Eighteenth International Workshop on Data Warehousing and OLAP*. 2015. ACM.

A 112. Nikaein, N., et al. *Network store: Exploring slicing in future 5G networks*. ACM.

A 113. Zhao, M., et al., *Multi-level VM Replication based Survivability for Mission-critical Cloud Computing.*

A 114. Rego, P.A.L., et al., *Using Processing Features for Allocation of Virtual Machines in Cloud Computing.* IEEE Latin America Transactions, 2015. **13**(8): p. 2798-2812.

A 115. Cheng, Y. and F. Rusu, *SCANRAW: A database meta-operator for parallel in-situ processing and loading.* ACM Transactions on Database Systems (TODS), 2015. **40**(3): p. 19.

A 116. Wang, K., et al., *Characterizing the impact of the workload on the value of dynamic resizing in data centers.* Performance Evaluation, 2015. **85-86**: p. 1-18.

A 117. Cui, L., et al., *PARS: A Page-Aware Replication System for Efficiently Storing Virtual Machine Snapshots.* SIGPLAN Not., 2015. **50**(7): p. 215-228.

A 118. Abdelaziz, E. and O. Mohamed. *Optimisation of the queries execution plan in cloud data warehouses*. in *2015 5th World Congress on Information and Communication Technologies (WICT)*. 2015. IEEE.

A 119. Bhagwat, D., et al., *A practical implementation of clustered fault tolerant write acceleration in a virtualized environment*, in *Proceedings of the 13th USENIX Conference on File and Storage Technologies*. 2015, USENIX Association: Santa Clara, CA. p. 287-300.

A 120. Ganesh Chandra, D., *BASE analysis of NoSQL database.* Future Generation Computer Systems, 2015. **52**: p. 13-21.

A 121. Ogunde, A., O. Folorunso, and A. Sodiya, *A partition enhanced mining algorithm for distributed association rule mining systems.* Egyptian Informatics Journal, 2015. **16**(3): p. 297-307.

A 122. Kavvadia, E., et al., *Elastic virtual machine placement in cloud computing network environments.* Computer Networks, 2015. **93**: p. 435-447.

A 123. Hussein, A., et al. *Impact of GC design on power and performance for Android*. in *Proceedings of the 8th ACM International Systems and Storage Conference*. 2015. ACM.

A 124. Shelar, M., S. Sane, and V. Kharat, *Enhancing Performance of Applications in Cloud using Hybrid Scaling Technique.* International Journal of Computer Applications, 2016. **975**: p. 8887.

A 125. Choyi, V.K., et al. *Network slice selection, assignment and routing within 5G networks*. IEEE.

A 126. He, M., et al., *Reverse replication of virtual machines (rRVM) for low latency and high availability services*, in *Proceedings of the 9th International Conference on Utility and Cloud Computing*. 2016, ACM: Shanghai, China. p. 118-127.

A 127. Sheikhi, A., M. Rayati, and A.M. Ranjbar, *Dynamic load management for a residential customer; Reinforcement Learning approach.* Sustainable Cities and Society, 2016. **24**: p. 42-51.

A 128. Galaktionov, V., et al. *A study of several matrix-clustering vertical partitioning algorithms in a disk-based environment*. in *International Conference on Data Analytics and Management in Data Intensive Domains*. 2016. Springer.

A 129. Wang, B., et al. *Server consolidation for internet applications in virtualized data centers*. in *Proceedings of the 24th High Performance Computing Symposium*. 2016. Society for Computer Simulation International.

A 130. Farokhi, S., et al., *A hybrid cloud controller for vertical memory elasticity: A control-theoretic approach.* Future Generation Computer Systems, 2016. **65**: p. 57-72.

A 131. Choudhary, A., S. Rana, and K.J. Matahai, *A Critical Analysis of Energy Efficient Virtual Machine Placement Techniques and its Optimization in a Cloud Computing Environment.* Procedia Computer Science, 2016. **78**: p. 132-138.

A 132. Pan, X., et al. *HogMap: Using SDNs to incentivize collaborative security monitoring*. ACM.

A 133. Seybold, D., et al. *Is elasticity of scalable databases a myth?* in *2016 IEEE International Conference on Big Data (Big Data)*. 2016. IEEE.

A 134. Sotiriadis, S., et al., *Vertical and horizontal elasticity for dynamic virtual machine reconfiguration.* IEEE Transactions on Services Computing, 2016(99): p. 1-1.

A 135. Meng, Y., et al. *CRUPA: A container resource utilization prediction algorithm for auto-scaling based on time series analysis*. IEEE.

A 136. Alharbi, Y. and S. Walker. *Data intensive, computing and network aware (dcn) cloud vms scheduling algorithm*. in *2016 Future Technologies Conference (FTC)*. 2016. IEEE.

A 137. Chung, Y., K.-b. Song, and K.-s. Cho. *A preemptible resource management scheme on multimedia processing cloud systems*. in *2016 International Conference on Information and Communication Technology Convergence (ICTC)*. 2016. IEEE.

A 138. Tanaka, T., et al., *Multiperiod IP-over-elastic network reconfiguration with adaptive bandwidth resizing and modulation.* Journal of Optical Communications and Networking, 2016. **8**(7): p. A180-A190.

A 139. Amato, A., et al. *Cossmic smart grid migration in federated clouds*. IEEE.

A 140. Moltó, G., M. Caballer, and C. de Alfonso, *Automatic memory-based vertical elasticity and oversubscription on cloud platforms.* Future Generation Computer Systems, 2016. **56**: p. 1-10.

A 141. Vaidya, M. and S. Deshpande, *Critical Study of Performance Parameters on Distributed File Systems Using MapReduce.* Procedia Computer Science, 2016. **78**: p. 224-232.

A 142. Huang, G., et al. *Auto scaling virtual machines for web applications with queueing theory*. in *2016 3rd International Conference on Systems and Informatics (ICSAI)*. 2016. IEEE.

A 143. de Assunção, M.D., et al., *Impact of user patience on auto-scaling resource capacity for cloud services.* Future Generation Computer Systems, 2016. **55**: p. 41-50.

A 144. Pietri, I. and R. Sakellariou, *Mapping virtual machines onto physical machines in cloud computing: A survey.* ACM Computing Surveys (CSUR), 2016. **49**(3): p. 49.

A 145. Marcoullis, I., *Self-stabilizing Middleware Services*, in *Proceedings of the Doctoral Symposium of the 17th International Middleware Conference*. 2016, ACM: Trento, Italy. p. 1-4.

A 146. Hwang, K., et al., *Cloud performance modeling with benchmark evaluation of elastic scaling strategies.* IEEE Transactions on Parallel and Distributed Systems, 2016. **27**(1): p. 130-143.

A 147. Hamrouni, T., S. Slimani, and F.B. Charrada, *A survey of dynamic replication and replica selection strategies based on data mining techniques in data grids.* Engineering Applications of Artificial Intelligence, 2016. **48**: p. 140-158.

A 148. Sartakov, V.A. and R. Kapitza. *Multi-site Synchronous VM Replication for Persistent Systems with Asymmetric Read/Write Latencies*. in *2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC)*. 2017.

A 149. Nemati, H., S.D. Sharma, and M.R. Dagenais, *Fine-grained Nested Virtual Machine Performance Analysis Through First Level Hypervisor Tracing*, in *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. 2017, IEEE Press: Madrid, Spain. p. 84-89.

A 150. Huang, L., et al. *Computation partitioning for mobile cloud computing in a big data environment*. in *2017 2nd International Conference on Frontiers of Sensors Technologies (ICFST)*. 2017. IEEE.

A 151. Vondra, T. and J. Šedivý, *Cloud autoscaling simulation based on queueing network model.* Simulation Modelling Practice and Theory, 2017. **70**: p. 83-100.

A 152. Montoya, G., et al., *Decomposing federated queries in presence of replicated fragments.* Journal of Web Semantics, 2017. **42**: p. 1-18.

A 153. Lastra-Díaz, J.J., et al., *HESML: A scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset.* Information Systems, 2017. **66**: p. 97-118.

A 154. Meiklejohn, C.S., et al. *Practical evaluation of the lasp programming model at large scale: An experience report*. in *Proceedings of the 19th International Symposium on Principles and Practice of Declarative Programming*. 2017. ACM.

A 155. Costache, S., et al., *Market-based autonomous resource and application management in private clouds.* Journal of Parallel and Distributed Computing, 2017. **100**: p. 85-102.

A 156. Putra, J.P., S.M.S. Nugroho, and I. Pratomo. *Live migration based on cloud computing to increase load balancing*. in *2017 International Seminar on Intelligent Technology and Its Applications (ISITIA)*. 2017.

A 157. Seybold, D. and J. Domaschka. *Is Distributed Database Evaluation Cloud-Ready?* in *European Conference on Advances in Databases and Information Systems*. 2017. Springer.

A 158. Zhou, A., et al., *Cloud Service Reliability Enhancement via Virtual Machine Placement Optimization.* IEEE Transactions on Services Computing, 2017. **10**(6): p. 902-913.

A 159. Li, R., et al., *A replication strategy for a distributed high-speed caching system based on spatiotemporal access patterns of geospatial data.* Computers, Environment and Urban Systems, 2017. **61**: p. 163-171.

A 160. Mansouri, N., M.K. Rafsanjani, and M.M. Javidi, *DPRS: A dynamic popularity aware replication strategy with parallel download scheme in cloud environments.* Simulation Modelling Practice and Theory, 2017. **77**: p. 177-196.

A 161. Rost, P., et al., *Network slicing to enable scalability and flexibility in 5G mobile networks.* IEEE Communications magazine, 2017. **55**(5): p. 72-79.

A 162. Foukas, X., et al., *Network slicing in 5G: Survey and challenges.* IEEE Communications Magazine, 2017. **55**(5): p. 94-100.

A 163. AbdElfattah, E., M. Elkawkagy, and A. El-Sisi. *A reactive fault tolerance approach for cloud computing*. in *2017 13th International Computer Engineering Conference (ICENCO)*. 2017. IEEE.

A 164. Sciancalepore, V., F. Cirillo, and X. Costa-Perez. *Slice as a service (SlaaS) optimal IoT slice resources orchestration*. IEEE.

A 165. George, S. and E.B. Edwin. *A Review On Data Replication Strategy In Cloud Computing*. in *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. 2017. IEEE.

A 166. Milani, B.A. and N.J. Navimipour, *A Systematic Literature Review of the Data Replication Techniques in the Cloud Environments.* Big Data Research, 2017. **10**: p. 1-7.

A 167. Foukas, X., M.K. Marina, and K. Kontovasilis. *Orion: RAN slicing for a flexible and cost-effective multi-service mobile network architecture*. ACM.

A 168. Beck, M., W. Hao, and A. Campan. *Accelerating the Mobile cloud: Using Amazon Mobile Analytics and k-means clustering*. in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*. 2017.

A 169. Basu, S. and P.K. Pattnaik. *A consistency preservation based approach for data-intensive cloud computing environment*. in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. 2017. IEEE.

A 170. López, M.R. and J. Spillner. *Towards quantifiable boundaries for elastic horizontal scaling of microservices*. in *Companion Proceedings of the10th International Conference on Utility and Cloud Computing*. 2017. ACM.

A 171. Apolónia, N., F. Freitag, and L. Navarro, *Leveraging deployment models on low-resource devices for cloud services in community networks.* Simulation Modelling Practice and Theory, 2017. **77**: p. 390-406.

A 172. Wang, C., et al. *A Fast, General Storage Replication Protocol for Active-Active Virtual Machine Fault Tolerance*. in *2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS)*. 2017.

A 173. Lin, H., et al. *Research on building an innovative electric power marketing business application system based on cloud computing and microservices architecture technologies*. in *2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*. 2017. IEEE.

A 174. Al-Dhuraibi, Y., et al., *Elasticity in cloud computing: state of the art and research challenges.* IEEE Transactions on Services Computing, 2017. **11**(2): p. 430-447.

A 175. Rossi, F.D., et al. *Dynamic network bandwidth resizing for big data applications*. IEEE.

A 176. Toffetti, G., et al., *Self-managing cloud-native applications: Design, implementation, and experience.* Future Generation Computer Systems, 2017. **72**: p. 165-179.

A 177. Li, J., et al., *Computation partitioning for mobile cloud computing in a big data environment.* IEEE Transactions on Industrial Informatics, 2017. **13**(4): p. 2009-2018.

A 178. López-Pires, F., et al., *Virtual machine placement for elastic infrastructures in overbooked cloud computing datacenters under uncertainty.* Future Generation Computer Systems, 2018. **79**: p. 830-848.

A 179. Husain, S., et al. *Mobile edge computing with network resource slicing for internet-of-things*. IEEE.

A 180. Kaur, G., A. Bala, and I. Chana, *An intelligent regressive ensemble approach for predicting resource usage in cloud computing.* Journal of Parallel and Distributed Computing, 2019. **123**: p. 1-12.

A 181. Qu, C., R.N. Calheiros, and R. Buyya, *Auto-scaling web applications in clouds: A taxonomy and survey.* ACM Computing Surveys (CSUR), 2018. **51**(4): p. 73.

A 182. Freitas, L.A., et al. *Slicing and allocation of transformable resources for the deployment of multiple virtualized infrastructure managers (vims)*. IEEE.

A 183. Raluca, O. and P. Florin. *Energy-Efficient Virtual Machine Replication for Data Centers*. in *2018 17th International Symposium on Parallel and Distributed Computing (ISPDC)*. 2018.

A 184. J.V, B.B. and D. Dharma, *HAS: Hybrid auto-scaler for resource scaling in cloud environment.* Journal of Parallel and Distributed Computing, 2018. **120**: p. 1-15.

A 185. Babu, K.R.R. and P. Samuel, *Interference aware prediction mechanism for auto scaling in cloud.* Computers & Electrical Engineering, 2018. **69**: p. 351-363.

A 186. Moghaddam, S.K., R. Buyya, and K. Ramamohanarao, *ACAS: An anomaly-based cause aware auto-scaling framework for clouds.* Journal of Parallel and Distributed Computing, 2019. **126**: p. 107-120.

A 187. Iqbal, W., A. Erradi, and A. Mahmood, *Dynamic workload patterns prediction for proactive auto-scaling of web applications.* Journal of Network and Computer Applications, 2018. **124**: p. 94-107.

A 188. Nadjaran Toosi, A., et al., *ElasticSFC: Auto-scaling techniques for elastic service function chaining in network functions virtualization-based clouds.* Journal of Systems and Software, 2019. **152**: p. 108-119.

A 189. Naskos, A., A. Gounaris, and I. Konstantinou. *Elton: A Cloud Resource Scaling-Out Manager for NoSQL Databases*. in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. 2018. IEEE.

A 190. Vogt, M., A. Stiemer, and H. Schuldt. *Polypheny-DB: Towards a Distributed and Self-Adaptive Polystore*. in *2018 IEEE International Conference on Big Data (Big Data)*. 2018.

A 191. Garcia-Aviles, G., et al., *POSENS: a practical open source solution for end-to-end network slicing.* IEEE Wireless Communications, 2018. **25**(5): p. 30-37.

A 192. Alami Milani, B. and N. Jafari Navimipour, *A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions.* Journal of Network and Computer Applications, 2016. **64**: p. 229-238.

A 193. Breitgand, D., et al., *Dynamic virtual machine resizing in a cloud computing infrastructure*. 2018, Google Patents.

A 194. Cappanera, P., F. Paganelli, and F. Paradiso, *VNF placement for service chaining in a distributed cloud environment with multiple stakeholders.* Computer Communications, 2019. **133**: p. 24-40.

A 195. Azari, L., et al., *A data replication algorithm for groups of files in data grids.* Journal of Parallel and Distributed Computing, 2018. **113**: p. 115-126.

A 196. Chen, J., et al., *A periodicity-based parallel time series prediction algorithm in cloud computing environments.* Information Sciences, 2019. **496**: p. 506-537.

A 197. Albers, S. and J. Quedenfeld, *Optimal Algorithms for Right-Sizing Data Centers*, in *Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures*. 2018, ACM: Vienna, Austria. p. 363-372.

A 198. Challa, R., et al. *SuperFlex: Network slicing based super flexible 5G architecture*. ACM.