Analyzing Auto-scaling Issues in Cloud Environments

Hanieh Alipour, Yan Liu, Abdelwahab Hamou-Lhadj
Electrical and Computer Engineering department
Concordia University, Montreal, Quebec, Canada
h_alipou@encs.concordia.ca, {yan.liu, wahab.hamou-lhadj} @concordia.ca

Abstract

Cloud computing is becoming increasingly widespread and sophisticated. A key feature of cloud computing is elasticity, which allows the provisioning and de-provisioning of computing resources on demand, via auto-scaling. Auto-scaling techniques are diverse, and involve various components at the infrastructure, platform and software levels. Auto-scaling also overlaps with other quality attributes, thereby contributing to service level agreements, and often applies modeling and control techniques to make the auto-scaling process adaptive. A study of auto-scaling architectures, existing techniques and open issues provides a comprehensive understanding to identify future research solutions. In this paper, we present a survey that explores definitions of related concepts of auto-scaling and a taxonomy of auto-scaling techniques. Based on the survey results, we then outline open issues and future research directions for this important subject in cloud computing.

1 Introduction

Cloud computing is an emerging computing model. The National Institute of Standards and Technology [1] defines cloud computing as: *"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*."

With cloud computing, provisioned resources can be changed according to the fluctuating demands of the customer, thus avoiding resource under-utilization and over-utilization, while maintaining a high level of quality for the hosted service. This feature is called *elasticity*, and forms the basis of the utility computing model. Therefore, customers pay only when infrastructural resources are needed. Cloud computing is also beneficial from the cloud provider's point of view because more customers can be served with the same infrastructure.

Tools that automatically modify the amount of used resources are called "auto-scaling services". Although auto-scaling has shown considerable potential for cloud computing, it also brings unique challenges that need to be addressed.

- Lack of auto-scaling studies of at the service level. Auto-scaling includes diverse cloud service models, but most studies only focus on the infrastructure level. Auto-scaling at the service-level is important as services are running on a set of connected VMs, and the quality of the service relies on how auto-scaling handles resources for these VMs. The service level metrics such as transactions per unit time need to be mapped to system level metrics such as CPU usage, network and disk IO rates.
- Insufficient tools for monitoring and aggregating metrics at the platform level and service level to support auto-scaling decisions.
- Auto-scaling in hybrid cloud environments is not well supported. Hybrid clouds are where part of the application is deployed on a private cloud, and the other part on a public cloud. In this scenario, the public and private

cloud may offer different auto-scaling techniques that are not compatible with each other, so there would be an interoperability issue in auto-scaling resources across the two clouds.

- The efficiency of auto-scaling in term of the reliability of the auto-scaling process is not well managed. Failure of the auto-scaling process can result in violations of the system's QoS requirements of performance and scalability and even incur unnecessary cost.
- There is a lack of studies to show the relationship between auto-scaling and quality attributes such as availability, reliability and security. For example, DoS attacks can cause an auto-scaling service to scale out the system unnecessarily and thus increase operation cost.

In this paper, we present a survey of auto-scaling techniques and concepts, architectural principles, state-of-the-art implementations, and research challenges. The ultimate goal is to understand auto-scaling services and to identify future research directions. The survey includes a manual search of the literature on the topic of "auto-scaling in cloud computing", and from the literature we extrapolate the main issues and topics that the auto-scaling field faces today.

The remaining parts of the paper are structured as follows. In Section 2, we introduce the definition of auto-scaling as well as the definitions of other relevant terms. In Section 3, we present the review methods in our literature search. In Section 4, we highlight the auto-scaling taxonomy to categorize various aspects of auto-scaling services. In Section 5, we present a categorization of the literature, and in Section 6 we discuss the open issues and future direction of auto-scaling. Finally, we conclude the paper in Section 7.

2 Definition of Auto-scaling

We first introduce the concept of auto-scaling and then discuss how auto-scaling differs from related concepts, namely resource provisioning, scalability, and elasticity.

The concept of auto-scaling has been loosely defined from many perspectives by academics and cloud technology vendors in diverse contexts. Gartner defines auto-scaling as follows:

"Auto-scaling automates the expansion or contraction of system capacity that is available for applications and is a commonly desired feature in cloud IaaS and PaaS offerings. When feasible, technology buyers should use it to match provisioned capacity to application demand and save costs." [2]

In Amazon Web Service (AWS), auto-scaling is defined as a cloud computing service feature that allows AWS users to automatically launch or terminate virtual instances based on defined policies, health status checks, and schedules [3].

Meanwhile, In RightScale [4], auto-scaling is defined as "*a way to automatically scale up or down the number of compute resources that are being allocated to your application based on its needs at any given time."*

From an academic point of view, auto-scaling is the capability in cloud computing infrastructures that allows dynamic provisioning of virtualized resources [5, 6]. Resources used by cloud-based applications can be automatically increased or decreased, thereby adapting resource usage to the applications' requirements [5].

Based on these definitions, the key features of auto-scaling are:

- The ability to scale out (i.e., the automatic addition of extra resources during increased demand) and scale in (i.e., the automatic termination of extra unused resources when demand decreases, in order to minimize cost).
- The capability of setting rules for scaling out and in.
- The facility to automatically detect and replace unhealthy or unreachable instances.

Auto-scaling is often referred in the context of resource provisioning, scalability, and elasticity. These terms are often used interchangeably, but they are actually slightly different concepts. Understanding the differences between these concepts can help us to identify the unique issues of auto-scaling and focus on the solutions.

Resource provisioning allows a system to scale out and in resources under dynamic workload [7, 8]. Efficient resource provisioning leads to improved scalability. Scalability enables a system to maintain performance during an increased workload by the addition of hardware resources [6], mostly by the system's administrator. There are two types of scalability: *horizontal scaling*, also known as scaling out, increases resources by

adding nodes or machines to the system. Meanwhile, *vertical scaling*, also known as scaling up, increases resources, such as CPU and processing power, in existing nodes.

Herbst et al. [6] explain the relation between scalability and elasticity as "*Scalability is a prerequisite for elasticity, but it does not consider temporal aspects of how fast, how often, and at what granularity scaling actions can be performed. Scalability is the ability of the system to sustain increasing workloads by making use of additional resources, and therefore, in contrast to elasticity, it is not directly related to how well the actual resource demands are matched by the provisioned resources at any point in time*."

Elasticity is defined as, "*The degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible*" [6]. In other words, the term elasticity covers how quickly the system can respond to fluctuating demands. Thus, auto-scaling techniques enable elasticity.

3 Related Work

Many survey papers have already covered the field of cloud computing. Ahmed et al. [103] provided a survey on cloud computing and state of the art research issues. The goal of this survey is to provide a general overview and better understanding of cloud computing, and as such, it is not specific to auto-scaling. Furthermore, the taxonomy in [104] illustrates the open issues related to cloud computing and describe a comprehensive study of cloud computing services. Zhang et al. [9] presented the fundamental concepts of cloud computing, the architectural designs, key technologies and research directions. Aceto et al. [10] focused on key properties and issues of cloud monitoring. Lorido-Botran et al. [11] focused on the current issues of auto-scaling and provided a category of auto-scaling techniques into five aspects, namely static, threshold-based polices, reinforcement learning, queueing theory, control theory and time-series analysis. Their survey, however, is limited to auto-scaling techniques from the IaaS's client perspective. Auto-scaling related to IaaS management, PaaS, and SaaS was not covered.

To the best of our knowledge, these surveys still lack a detailed analysis of auto-scaling for the Cloud. To fill this gap, we provide a survey on auto-scaling driven by research questions and a careful analysis and categorization of auto-scaling systems for the Cloud, the issues arising from these systems and how such issues have been tackled in the literature.

4 The Review Method

To collect papers in the literature, we focus on the following research questions:

- What is the relationship between auto-scaling and other cloud functionalities, such as monitoring and quality of service management?
- How auto-scaling correlate to other quality does attributes including performance, scalability, availability, and dependability?
- How does auto-scaling impose technical issues to different domain applications, such as video streaming, databases, health care, and mobile applications?

Driven by these research questions, we first searched papers that covered auto-scaling issues in cloud computing conference proceedings and journal papers, using the following well-known online libraries:

- ACM Digital Library (http://dl.acm.org/)
- Google Scholar (http://scholar.google.com)
- IEEE Xplore (http://ieeexplore.ieee.org)
- ScienceDirect (http://www.sciencedirect.com)
- SpringerLink (http://www.springer.com)

Our initial search included papers that were cited in surveys related to auto-scaling in cloud computing. In order to judge the relevancy of the papers to our topic, we reviewed their title, abstract, introduction, and approach sections.

A) Inclusion criteria:

Our inclusion criteria were for papers that addressed the problems in this field (problem domain) and those that proposed solutions to these problems (solution domain). In addition, we included papers that were not specifically on auto-scaling, but their topics were intrinsically linked to auto-scaling. For example, we included papers on monitoring and security, because they addressed crosscutting issues with auto-scaling. That is not to say that we included all topics on monitoring - a search of the keyword "monitoring" produces over 200 papers alone, of which most

were irrelevant to our search. To ensure the quality of the papers, we only included peer-reviewed papers.

B) Exclusion criteria:

We excluded papers not written in English. We also did not include reports, pamphlets and reviews in our search. We also did not include a paper if it did not provide validation for the proposed solution.

C) Data extraction:

In order to answer the research questions, our next step was to extract the information from the papers. To do this in a systematic way, we define taxonomy to categorize the topics covered in the selected literature.

5 Taxonomy

Concentrated on the three research questions listed in Section 4, we have categorized the literature into five main topics, which are: 1) *Level of auto-scaling in Cloud*, 2) *Quality attributes and crosscutting concerns*, 3) *Domains and applications, 4) Affiliated management*, and *5) Modeling and prediction*.

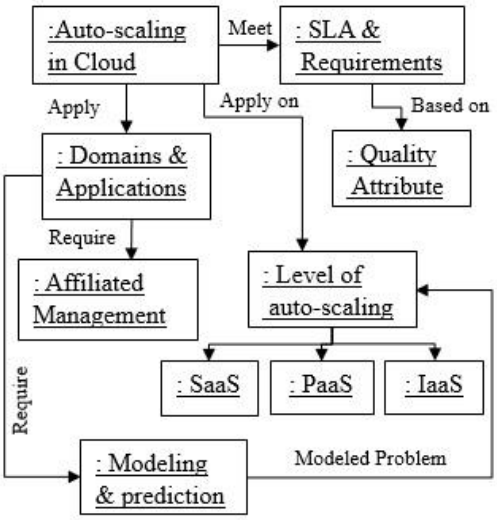

Figure 1: Conceptual diagram

These five topics are intrinsic to each other, as depicted in Figure 1. Each topic is further divided into smaller subtopics, which form the taxonomy shown in Figure 2.

5.1 Level of Auto-scaling

One key function of auto-scaling is to guarantee Service Level Agreements (SLAs) from the service provider side, and to satisfy requirements for quality attributes (e.g., scalability, availability, and reliability) from the perspective of cloud applications. Driven by SLAs and requirements for quality attributes, auto-scaling is applicable to three different cloud service models: SaaS, PaaS, and IaaS.

A) Auto-scaling at the IaaS level

Auto-scaling at the IaaS (Infrastructure-as-a-Service) level involves two main groups: applying proprietary vendor techniques and devising auto-scaling mechanisms for specific needs.

In the former group, Amazon Web Service (AWS) is a powerful public cloud provider with its own auto-scaling mechanism. One example of a service using AWS is YinzCam, which is a cloud-hosted service that provides sports information to sports fans. Mickulicz et al. [12] have discussed the limitations of the original YinzCam architecture, and the move of the system to AWS. Mickulicz et al. showed how auto-scaling can hide architecture inefficiencies, but at higher operational costs. Cloud-hosted applications have fluctuating workloads but also need to fulfill SLAs, therefore cloud-hosted applications require resource provisioning to be SLA-aware. Thus, it is essential to obtain a good understanding of the performance behavior of virtual instances. Dejun et al. [13] analyzed the resource provisioning performance of the Amazon Elastic Compute Cloud (Amazon EC2) in service-oriented architecture. In their paper, they calculated the performance stability and consistency of small instances in the Amazon EC2.

The topic on devising *auto-scaling mechanisms* consists of subtopics that address specific needs and techniques of auto-scaling including virtualization, comparison, workload monitoring, hybrid/multi-clouds, bandwidth, integrated storage, network and computing, self-scaling frameworks.

Virtualization is perhaps the most addressed topic; many studies use virtualization to design and propose new auto-scaling mechanisms [14, 15, 46, 90, 91, 73, 8, 74, 20, 72, 51], including VM allocation [14, 15, 46, 90, 91, 73], tuning VM capacity [8, 74], DoS attack issue (security issue) [20], and elastic VM architecture [72, 51].

A cloud auto-scaling mechanism presented in [15] scheduled VM instance fire up or turn off activities, automatically scaling the instances based on workload information and performance desire.

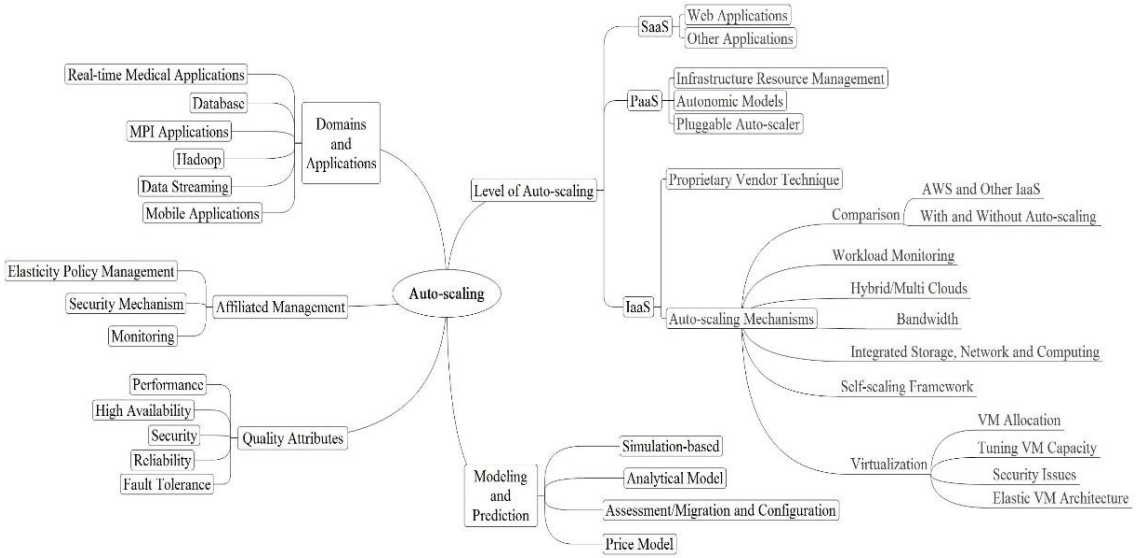

Figure 2: Taxonomy of auto-scaling:

Erdil [16] worked on inter-cloud federations. He proposed proxies that broadcast information to improve the success of distributed cloud resource schedulers. MODAClouds [17] is a model-driven approach for the design and execution of applications on multiple clouds to support developers migrating between clouds.

Ahn et al. [14] investigated the weaknesses of existing scaling mechanisms in real-time healthcare applications. They proposed new auto-scaling mechanisms in order to dynamically adjust the number of VMs. They combined an independent real-time resource monitor, a virtual session manager, and a workload prediction algorithm to make the mechanisms reliable and efficient.

B) Auto-scaling at the PaaS level

At the PaaS (Platform-as-a-Service) level, auto-scaling mainly deals with three subtopics, namely, *infrastructure* resource management, autonomic models for PaaS, and pluggable auto-scaler in PaaS.

The reason we propose infrastructure resource management as a subtopic is that cloud applications at the PaaS level share and compete for resources simultaneously. Hence, the issue of integrating and coordinating the resource consumption and allocation is pertinent at the PaaS level. Zhang et al. [64] integrated resource consumption of a PaaS application and provisioning decisions using a control-loop based approach.

An autonomic model for PaaS refers to a model with the ability to self-monitor, self-repair, and self-optimize. To serve this purpose, Buyya et al. [71] proposed an architecture for developing self-governed resource provisioning and management techniques in PaaS.

Pluggable auto-scalar in PaaS means a service that extends the PaaS level to allow the user to experiment with existing or new extensions. Bunch et al. [18] designed and implemented an open-source, pluggable auto-scaling service for PaaS systems that runs at the cloud PaaS layer. Their work also contains high availability-awareness, as well as QoS-awareness.

C) Auto-scaling at the SaaS level

At the SaaS (Software-as-a-Service) level, current cloud service providers offer services based on a pay-per-use business model. Issues such as managing resources according to fluctuating loads in SaaS applications, and meeting users' expectations of Quality-of-Service (QoS) have become attractive for researchers [7, 19, 49, 79, 80, 81].

Many studies focus on web applications. We divided SaaS studies into those involving web applications [49, 80, 7 and 19] and those that concentrate on other types of applications, such as Hadoop and parallel applications [79, 81].

One study that deals with web applications at the SaaS level is on solving resource overutilization of multi-tenancy applications [19]. *Multi-tenancy* refers to the ability to offer one single application instance to several tenants. An SaaS platform and its applications should be aware of how tenants use resources. Current cloud virtualization mechanisms do not provide cost-effective pay-per-use models for SaaS applications and just-in-time scalability is not achieved by simply deploying SaaS applications to cloud platforms [19]. Espadas et al. [19] proposed a tenant-based resource allocation model that consists of three approaches: (1) Tenant-based isolation to encapsulate the execution of each tenant; (2) Tenant-based load balancing to distribute requests; and (3) Tenant-based VM allocation to verify the number of VM instances which are needed for a certain workload.

5.2 Quality attributes

Quality attributes affect run-time behavior, user experience, and system design. Some quality attributes (such as performance, security, high availability, reliability and fault tolerance) are strongly related to auto-scaling in cloud environments. These quality attributes play an important role for new models of quantifying auto-scaling problems and the design of auto-scaling mechanisms. One example of achieving performance through auto-scaling is the video-on-demand (VoD) service with stringent bandwidth requirements to guarantee the performance of VoD services. Niu et al. [93] proposed a predictive resource auto-scaling system that dynamically reserves the minimum bandwidth resources from multiple data centers for the VoD provider.

Ferraris et al. [53] evaluated the auto-scaling performance of the Flexiscale [105] and Amazon EC2 cloud hosting platforms. Their result demonstrated the difficultly of changing auto-scaling parameters, such as the minimum and maximum pool size, and the scale in and scale out thresholds. They suggested implementing a pro-active behavior, since simple actions taken after threshold violations are not enough to guarantee performance.

Bunch et al [18] designed an auto-scaler that runs at the PaaS layer. They designed a role-based approach. Each role indicated the responsibility of the node, and when it should be started and stopped. The auto-scaler could view metrics and roles of nodes, and was responsible for making decisions to start or stop nodes. They also contributed cost-aware auto-scaler, which focuses on saving costs rather than QoS. The authors claimed that their approach saved applications 91% of the instances they normally use in the PaaS, although with a lower QoS.

5.3 Domains and applications

Mobile applications [21], database systems [22,100], data stream applications [79], Hadoop and real-time medical applications [102, 56] are typical applications that have the most need for data intensive computing and on-demand resource provisioning. These applications have fluctuating loads and apply auto-scaling mechanisms to handle resource provisioning.

Nowadays, clients expect to use cloud applications on a variety of smart devices. Bernstein et al. [21] gave an example of a car driver using a GPS mobile application running on the cloud. Bernstein et al. presented a platform architecture to support capacity changes. However, auto-scaling was not featured in their work. Huang et al. [100] proposed an auto-scaling database virtualization approach to satisfy SLA requirements. They combined the auto-scaling feature of clouds with the sharing features of MongoDB to create a rapid auto-scaling cloud storage system.

Vijayakumar et al. [79] focused on auto-scaling problems for data stream applications in clouds. In data stream applications, external sources generate data. The authors proposed an approach to match the processing rate with the data arrival rate by cautiously allocating resources. Their proposed algorithm handled dynamic patterns of data arrivals, while preventing degradation in processing rate.

5.4 Affiliated management

Affiliated management refers to cloud computing control mechanisms that are highly related to auto-scaling. The topic consists of three subtopics: Elastic policy management [54, 83, and 96], Monitoring [31, 101, 57, 23, 24, 10, 77], and Security mechanisms [20].

Elastic policy management is a vital part of auto-scaling systems. An elastic policy governs when and how resources are added or removed from a cloud environment. Elastic policy rules need to be optimized to reduce operation cost. However, determining a suitable policy can be challenging.

Another affiliated management of auto-scaling is monitoring. Metrics, at the infrastructure and platform levels, must be monitored so that an auto-scaling mechanism is aware of system's states.

Finally, security mechanisms are affiliated with auto-scaling since malicious attacks can exploit poorly designed auto-scaling systems. Ristenpart et al. [106] discussed the risks raised from sharing physical infrastructures, even when their actions were isolated through VMs, such as inside a third-party cloud service (e.g., Amazon EC2) in auto-scaling systems. They presented approaches to alleviate this risk. First, cloud providers may conceal or obfuscate the placement policy and the internal structure of their services. Secondly, they may employ masking techniques to minimize the information that can be leaked. Finally, they could just simply not allow users to share infrastructure with each other.

The effectiveness of auto-scaling has a direct impact on performance, scalability and availability. Thus, affiliated management for auto-scaling should provide the following features:

A) Awareness of the system's state and related entities in the environment.

The next generation of cloud computing integrates SaaS, PaaS and IaaS, and merges private and public clouds and cloud federations [23]. Smit et al. [23] considered the integration of private and public clouds and defined requirements for collecting monitoring data from different groups of resources. The cloud monitoring framework tracked resource consumption and cost in near real -time. DARGOS [77] is a distributed architecture for resource management and monitoring in clouds. It is responsible for publishing resource monitoring information, and is able to measure physical and virtual resources accurately, while ensuring a low overload. Katsaros et al. [24] proposed a monitoring system for measuring QoS at both the application and infrastructure levels, targeting trigger events for runtime adaptability of resource provisioning estimation and decision making.

B) Timely capturing the scaling pattern, which reduces risk of missing spike workload. Mickulicz et al. [12] observed that an AWS auto-scaling policy, such as scaling out if average CPU-usage is above 30% during one minute could cope with unpredictable spiky workloads to ensure a responsive user experience.

C) Responsiveness of the entity performing scaling-in and scaling-out cloud operations. The scaling entities should respond quickly to demands on resources. Slow operations could miss the best opportunity to catch up the workload and cause SLA violations.

D) Cost effectiveness.

Methods to optimize resources for auto-scaling in cloud computing include just-in time infrastructure, efficient resource utilization and usage-based pricing. There are a number of studies that present efficient price models. Mishra et al. [25] used shared-disk architecture to propose an optimal pricing solution, because cost efficiency of auto-scaling in cloud computing depends on a high workload density, a dynamic management of resources, an economy of scale and the ability to run in various environments. Cloud resources are charged in an hourly manner. Cloud providers now offer diverse instance types, at different prices. Choosing suitable instance types based on the application workload can further save money and improve performance. Mao et al [15] considered choosing cost-effective instance types to propose a cost-effective auto-scaling mechanism.

E) Appropriate handling of the failure of scaling operations.

Auto-scaling systems can fail because of unexpected node failures or other uncertainties. If a scaling system fails, the system will not be able to scale out or increase resources to match demands. The system should have a recovery process, so that in the event of a failure, the auto-scaling process is embedded into the fault-tolerant process to make auto-scaling reliable.

5.5 Modeling and prediction

Modeling and prediction techniques facilitate the management of auto-scaling by quantifying auto-scaling features and providing estimation of the performance and cost of a specific auto-scaling mechanism or architecture.

A) Simulation-based modeling and prediction

Simulation benefits cloud customers as it allows cloud clients to their services in a repeatable and controllable environment and adjust the auto-scaling configurations before deploying services to the cloud. There are two challenging problems with simulation-based studies, namely measuring the performance of resource allocation for different cloud applications, and modelling services under fluctuating loads.

Buyya et al. [27] offered a solution for evaluating different kinds of auto-scaling scenarios by proposing a simulation toolkit called "CloudSim" that modeled and simulated multiple data centers to scale out applications.

Thepparat et al. [28] simulated auto-scaling problems using virtualization technology, with the objective of supporting heavy workloads at peak times. The auto-scaling without server virtualization and auto-scaling with server virtualization were built on the ARENA simulation software. They claimed that the mean time of failure and CPU utilization could increase with server virtualization technology in auto-scaling.

B) Analytical models

An analytical model is a mathematical model that formulates the behavior of auto-scaling systems. Mao et al. [73] described a provisioning method that automatically adjusts to workload changes. Their work is based on a monitor-control loop that adjusts to dynamic changes such as the workload bursting and delayed instance acquisitions.

In [29], the resource provisioning problem was formulated in a two-phase algorithm. The first phase focused on providing optimal long term resources by proposing a mathematical formulae. The second phase proposed a Kalman filter prediction model to predict demand.

A novel framework proposed in [30] supported reactive and proactive approaches for implementing auto-scaling services. With a reactive approach, resources are scaled in or out in response to fluctuating user demands. With a proactive approach, on the other hand, future demand is predicted. Resources are scaled in advance for the increased or decreased demand. The authors developed a set of predictors for future demand on infrastructure resources, as well as a selection mechanism to choose the best predictor. The proactive approach was more successful at minimizing cost and SLO violations, while the reactive approach was useful for reducing resources that had already been over-provisioned. In their work, a scaling decision was taken proactively every five minutes, and a reactive decision was made whenever an action was activated.

Another study [31] introduced a new open source solution for auto-scaling, which focused on SLA compliance. Their work followed the MAPE loop (Monitoring, Analysis, Planning, Execution), in an industrial context.

Chazalet et al. [33] proposed an SLA-based multi-dimensional resource allocation schema in multi-tier applications. Their aim was to optimize the total profit gained under SLA contracts. Their solution focused on the capacity constraint of servers between different clients and a resource consolidation technique based on the force-direction search. A similar study, conducted by Goudarzi et al. [35], worked on multi-dimensional SLA-based resource allocation problem. The authors modelled the response time based on different resource allocations to increase profits.

Roy et al. [34] focused on optimizing resource allocation. In their paper, the authors discussed challenges in auto-scaling and shortages in techniques in workload forecasting. The authors presented a resource allocation algorithm based on a control algorithm to predict future workload used for auto-scaling. Their result demonstrated the model help to satisfy QoS requirements while remaining low operational costs.

Song et al. [36] drew attention to the fact that large data centers, such as Google, have auto-scaling systems that focused only on either local scaling within a server, or central global scaling. The authors thus proposed a two-tiered system, one that combines local and global resource allocation. Their system optimized resource allocation by preferentially giving resources to critical applications. Caron et al. [42] presented a solution for workload prediction by identifying similar past incidences of the current short-term workload history. The authors proposed a novel usage prediction algorithm for auto-scaling which use historic data to detect similar usage patterns and use it in scaling decision.

Yanggratoke et al. [43] addressed the problem of resource management for large-scale cloud with objective of saving a dynamic workload with minimal power consumption. The authors proposed a GRMP-Q (Generic protocol for Resource allocation for Minimizing Power consumption). The protocol provides a judicious allocation of CPU resources to clients.

Elastic Application Container (EAC) is a lightweight resource management model proposed by He et al. [37]. The proposed EAC-based resource management solution performed better than the VM-based solution. Hung et al. [38] considered energy cost to propose an auto-scaling algorithm for dynamic and balancing resource provisioning.

C) Assessment, Migration and *Configuration*

Legacy applications sometimes have to be migrated, but this can be complicated and costly if the application is limited to specific providers, or if migration is not an automated process. [39]. Frey et al. [39] proposed a tool called CloudMIG, which is a model-based approach for migrating software systems from SaaS providers towards public IaaS and PaaS-based clouds. The tool uses a model-driven approach to generate considerable parts of an architecture utilizing rule-based heuristics. Feedback loops allow for further alignment with the specific cloud environment's properties and improve resource efficiency and scalability. A model-driven engineering approach in [40, 41] is used to develop the Smart Cloud Optimization of Resource Configuration Handling (SCORCH) tool. The tool is built using feature models to maximize efficiency of auto-scaling queues and avoid boot-time penalties, thus considerably reduced allocation time.

Level	Sub-topics	
SaaS	Web Applications [49,80,7,19]	
	Other Applications [79,81]	
PaaS	Infrastructure Resource Management [64]	
	Autonomic Models [48,21,71]	
	Pluggable Auto-scaler [18]	
IaaS	Comparison [103,53,28]	
	Virtualization	VM Allocation [14,15,46,90,91,7]
		Tuning VM Capacity[8,74]
		Security Issues [20]
		Elastic VM Architecture [72,51]
	Monitoring [75,69]	
	Hybrid/multi-clouds [16,17,95,84,44,36,78,67,65]	
	Bandwidth [93]	
	Integrated Storage, Network and Computing [63]	
	Self-scaling Framework[49,50]	

Table 1: Level of Auto-scaling Topics

Sub-topics
Simulation-based [27,30]
Analytical Model-based [48,81,58,97,102,31,29,100,83,66,59,92,94,89, 70,55,35,34,76,32,33,38,37,36,20,49,52,82,85]
Assessment and Configuration [40,41,47,5,98,39,61, 86,42, 43,60]
Price Model [25,25,62,87,8]

Table 2: Modeling and Prediction Topic

5.6 Categorization

Tables 1, 2 and 3 provide classifications of the papers that are related to the topics of the taxonomy.

Topics	Sub-topics
Quality Attributes	Performance[93,53,30]
	Security[20,71]
	HA[101,99]
	Reliability [71]
Affiliated Management	Elastic policy configuration [54,83,96]
	Security mechanism [106]
	Monitoring [31,101,57,23,24,10,77]
Domains /Applications	Mobile Apps [21]
	Database [22,100]
	Data Streaming [79]
	Hadoop [102]
	MPI Apps [56]
	Real-time Medical Apps [14,72]
	Back-end Mashup [68]

Table 3: Quality Attribute, Affiliated

6 Open Issues

A) Migrating from one cloud to another cloud

Cloud providers usually offer similar services to clients. Auto-scaling is often supported by a cloud provider to allow clients to configure

the best resource scaling options to achieve their economic goals. A major problem is "vender-lock", where the service cannot be easily transferred to a competitor. Cloud providers are heterogeneous and their services are often incompatible. This prevents interoperability and increases the cost and complexity of migration to other clouds. There is a need for a mechanism that supports auto-scaling of resources, portability, interoperability and federation between clouds [16, 17].

Most of the research on auto-scaling focuses on a single cloud environment. Only a few studies, mentioned in Section 5, focused on multi-clouds [44].

B) Formulating the problem of optimizing the cost and configuration in dynamic resource allocation

Most existing auto-scaling techniques only consider system level resource utilization. There is a lack of consideration of SLAs, user performance requirements, and cost concerns. SLAs bring their own challenges: over-provisioning is costly whereas under-provisioning impairs performance. Auto-scaling needs to provide resources in response to unexpected load changes rapidly and on demand. However, existing mechanisms have delays of usually several minutes when allocating resources. Such a delay may lead to SLA violations for real-time services (such as Video on Demand services). On the other hand, maintaining idle resources incurs unnecessary operational expense. Optimizing the cost and configuration in dynamic resource allocation, while avoiding SLA violations, is an important open issue in cloud computing.

C) Auto-scaling, Monitoring tools and Cloud configurations

Monitoring tools in auto-scaling systems are essential for automatic resource provisioning. Users have a wide variety of monitoring tools to choose from, with each provider typically having their own set of monitoring tools. Usually, each tool solves one specific auto-scaling monitoring problem. Thus, an open issue is that the metrics collected from each monitoring tool need to be transformed and aggregated to fit the auto-scaling analysis. A further open issue is that most monitoring tools are for the infrastructure level, and there is a lack of tools for the platform and service levels.

D) Auto-scaling failures

Auto-scaling system failures are not well addressed. The auto-scaling process is subject to faults and failures from software, networking and hardware aspects. One scenario is that a certain number of nodes are needed, but only partial nodes are actually launched. When failures like this occur, an auto-scaling mechanism needs to recover in an intelligent way. Simply re-running the auto-scaling action may result in extra nodes being provisioned unnecessarily.

7 Conclusion

In recent years, cloud computing has attracted an increasing amount of attention from industry and academia alike. This is mainly due to cloud computing's ability to dynamically provision resources on-demand. The objective of this paper is to present a comprehensive study about the auto-scaling mechanisms available today, as well as to highlight the open issues in the field. In this paper, we provide a careful analysis of current state of auto-scaling in cloud computing. We first put auto-scaling into context by providing background information, discussing the main beneficiaries of auto-scaling, and giving definitions of key concepts. Next, we proposed a taxonomy that simplifies the state of auto-scaling today, and which provides researchers and developers with ideas about the current auto-scaling mechanisms and challenges. We then examined past and existing issues, and the contributions provided in literature so far. We then described the main platforms (commercial and academic), and finally, we considered the challenges and future directions of auto-scaling in cloud computing.

References

[1] NIST: http://www.nist.gov/itl/cloud/.

[2] Gartner: http://www.gartner.com/technology/home.jsp.

[3] Amazon Web Services, Inc.: //aws.amazon.com/documentation/autoscaling/.

[4] RightScale: http://support.rightscale.com/06-

FAQs/FAQ_0043_-_What_is_autoscaling%3F

[5] N. M. Calcavecchia, B. A. Caprarescu, E. Di Nitto, D. J. Dubois, and D. Petcu, "Depas: A decentralized probabilistic algorithm for auto-scaling," *Computing*, vol. 94, no. 8–10.

[6] N. R. Herbst, S. Kounev, and R. Reussner, "Elasticity in Cloud Computing: What It Is, and What It Is Not," in *Proceedings of the 10th International Conference on Autonomic Computing (ICAC 2013), San Jose, CA*, 2013.

[7] W. Iqbal, M. N. Dailey, D. Carrera, and P. Janecek, "Adaptive resource provisioning for read intensive multi-tier applications in the cloud," *Future Gener. Comput. Syst.*, vol. 27, no. 6, 2011.

[8] R. Han, L. Guo, M. M. Ghanem, and Y. Guo, "Lightweight resource scaling for cloud applications," in *Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on*, 2012, pp. 644–651.

[9] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *J. Internet Serv. Appl.*, vol. 1, no. 1, 2010.

[10] G. Aceto, A. Botta, W. De Donato, and A. Pescapè, "Cloud monitoring: A survey," *Comput. Netw*, vol. 57, no. 9, pp. 2093–2115, 2013

[11] T. Lorido-Botrán, J. Miguel-Alonso, and J. A. Lozano, "Auto-scaling techniques for elastic applications in cloud environments," *Dep. Comput. Archit. Technol. Univ. Basque Ctry. Tech Rep EHU-KAT-IK-09-12*, 2012.

[12] N. Mickulicz, P. Narasimhan, and R. Gandhi, "To Auto Scale or not to Auto Scale," in *Workshop on Management of Big Data Systems*, 2013.

[13] 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.*

[14] Y. W. Ahn, A. M. Cheng, 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 Netw.*, 2013.

[15] M. Mao, J. Li, and M. Humphrey, "Cloud auto-scaling with deadline and budget constraints," in *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on*, 2010.

[16] D. C. Erdil, "Autonomic cloud resource sharing for intercloud federations," *Future Gener. Comput. Syst.*, vol. 29, no. 7, 2013.

[17] D. Ardagna, E. Di Nitto, P. Mohagheghi, S. Mosser, C. Ballagny, F. D'Andria, G. Casale, P. Matthews, C.-S. Nechifor, and D. Petcu, "Modaclouds: A model-driven approach for the design and execution of applications on multiple clouds," in *Modeling in Software Engineering (MISE), 2012 ICSE Workshop on*, 2012.

[18] C. Bunch, V. Arora, N. Chohan, C. Krintz, S. Hegde, and A. Srivastava, "A pluggable autoscaling service for open cloud PaaS systems," in *Proceedings of the 2012 IEEE/ACM Fifth International Conference on Utility and Cloud.*

[19] J. Espadas, A. Molina, G. Jiménez, M. Molina, R. Ramírez, and D. Concha, "A tenant-based resource allocation model for scaling Software-as-a-Service applications over cloud computing infrastructures," *Future Gener. Comput. Syst.*, vol. 29, no. 1, pp. 273–286, 2013.

[20] A. Dolgikh, Z. Birnbaum, Y. Chen, and V. Skormin, "Behavioral Modeling for Suspicious Process Detection in Cloud Computing Environments," in *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*.

[21] D. Bernstein, N. Vidovic, and S. Modi, "A cloud PAAS for high scale, function, and velocity mobile applications-with reference application as the fully connected car," in *Systems and Networks Communications (ICSNC), 2010 Fifth International Conference on*, 2010, pp. 117–123.

[22] D. Agrawal, A. El Abbadi, S. Das, and A. J. Elmore, "Database scalability, elasticity, and autonomy in the cloud," in *Database Systems for Advanced Applications*, 2011, pp. 2–15.

[23] M. Smit, B. Simmons, and M. Litoiu, "Distributed, application-level monitoring for heterogeneous clouds using stream processing," *Future Gener. Comput. Syst.*, vol. 29, no. 8, 2013.

[24] G. Katsaros, G. Kousiouris, S. V. Gogouvitis, D. Kyriazis, A. Menychtas, and T. Varvarigou, "A Self-adaptive hierarchical monitoring mechanism for Clouds," *J. Syst. Softw.*, vol. 85, 2012.

[25] D. Mishra, N. Gaurha, and P. Trivedi, "Optimal Pricing Solution for Cloud Offered Business," in *Communication Systems and Network Technologies (CSNT), 2013 International Conference on*, 2013, pp. 365–370.

[26] Y. C. Lee, C. Wang, A. Y. Zomaya, and B. B. Zhou, "Profit-driven service request scheduling in clouds," in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, 2010, pp. 15–24.

[27] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities," in *High Performance Computing & Simulation, 2009. HPCS'09. International Conference on*, 2009, pp. 1–11.

[28] T. Thepparat, A. Harnprasarnkit, D. Thippayawong, V. Boonjing, and P. Chanvarasuth, "A Virtualization Approach to Auto-Scaling Problem.," in *ITNG*, 2011, pp. 169–173.

[29] R. Hwang, C. Lee, Y. Chen, and D. Zhang-Jian, "Cost Optimization of Elasticity Cloud Resource Subscription Policy," 2013.

[30] F. J. Almeida Morais, F. Vilar Brasileiro, R. Vigolvino Lopes, R. Araujo Santos, W. Satterfield, and L. Rosa, "Autoflex: Service agnostic auto-scaling framework for iaas deployment models," in *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium.*

[31] A. Chazalet, F. Dang Tran, M. Deslaugiers, F. Exertier, and J. Legrand, "Self-scaling the Cloud to meet Service Level Agreements," in *CLOUD COMPUTING 2010, The First International Conference on Cloud Computing, GRIDs, and Virtualization*, 2010.

[32] H. Ghanbari, B. Simmons, M. Litoiu, C. Barna, and G. Iszlai, "Optimal autoscaling in a IaaS cloud," in *Proceedings of the 9th international conference on Autonomic computing*, 2012,

[33] H. Goudarzi and M. Pedram, "Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, 2011, pp. 324–331.

[34] N. Roy, A. Dubey, and A. Gokhale, "Efficient autoscaling in the cloud using predictive models for workload forecasting," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, 2011, pp. 500–507.

[35] H. Goudarzi and M. Pedram, "Maximizing profit in cloud computing system via resource allocation," in *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on*, 2011, pp. 1–6.

[36] Y. Song, Y. Sun, and W. Shi, "A two-tiered on-demand resource allocation mechanism for VM-based data centers," *Serv. Comput. IEEE Trans. On*, vol. 6, no. 1, pp. 116–129, 2013.

[37] S. He, L. Guo, Y. Guo, C. Wu, M. Ghanem, and R. Han, "Elastic application container: A lightweight approach for cloud resource provisioning," in *Advanced information networking and applications (aina), 2012 ieee 26th international conference on*, 2012, pp. 15–22.

[38] C-L. Hung, Y-C. Hu, and K-C. Li, "Auto-Scaling Model for Cloud Computing System," *Int. J. Hybrid Inf. Technol.*, vol. 5, no. 2, 2012.

[39] S. Frey and W. Hasselbring, "Model-based migration of legacy software systems to scalable and resource-efficient cloud-based applications: The cloudmig approach," in *CLOUD COMPUTING 2010, The First International Conference on Cloud Computing, GRIDs, and Virtualization*, 2010.

[40] B. Dougherty, J. White, and D. C. Schmidt, "Model-driven auto-scaling of green cloud computing infrastructure," *Future Gener. Comput. Syst.*, vol. 28, no. 2, pp. 371–378, 2012.

[41] B. Dougherty, J. White, and D. C. Schmidt, "Model-driven Configuration of Cloud Computing Auto-scaling Infrastructure."

[42] E. Caron, F. Desprez, and A. Muresan, "Forecasting for grid and cloud computing on-demand resources based on pattern matching," in *Cloud Computing Technolo-*

gy and Science (CloudCom), 2010 IEEE Second International Conference on, 2010.

[43] R. Yanggratoke, F. Wuhib, and R. Stadler, "Gossip-based resource allocation for green computing in large clouds," in *Network and Service Management (CNSM), 2011 7th International Conference on*, 2011.

[44] N. Chondamrongkul and P. Temdee, "Multi-cloud computing platform support with model-driven application runtime framework," in *Communications and Information Technologies (ISCIT), 2013 13th International Symposium on*, 2013, pp. 715–719.

[45] OpenSAF Foundation: http://www.opensaf.org/.

[46] M. Mao and M. Humphrey, "Auto-scaling to minimize cost and meet application deadlines in cloud workflows," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, 2011.

[47] U. Sharma, P. Shenoy, S. Sahu, and A. Shaikh, "A cost-aware elasticity provisioning system for the cloud," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, 2011, pp. 559–570.

[48] R. Han, L. Guo, Y. Guo, and S. He, "A deployment platform for dynamically scaling applications in the cloud," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, 2011.

[49] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis, "Efficient resource provisioning in compute clouds via vm multiplexing," in *Proceedings of the 7th international conference on Autonomic computing*, 2010.

[50] J. Rao, X. Bu, C.-Z. Xu, and K. Wang, "A distributed self-learning approach for elastic provisioning of virtualized cloud resources," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on*, 2011.

[51] W. Dawoud, I. Takouna, and C. Meinel, "Elastic vm for cloud resources provisioning optimization," in *Advances in Computing and Communications*, Springer, 2011, pp. 431–445.

[52] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Gener. Comput. Syst.*, vol. 28, no. 1, pp. 155–162, 2012.

[53] F. L. Ferraris, D. Franceschelli, M. P. Gioiosa, D. Lucia, D. Ardagna, E. Di Nitto, and T. Sharif, "Evaluating the Auto Scaling Performance of Flexiscale and Amazon EC2 Clouds," in *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2012 14th International Symposium on*, 2012, pp. 423–429.

[54] H. Ghanbari, B. Simmons, M. Litoiu, and G. Iszlai, "Exploring alternative approaches to implement an elasticity policy," in *Cloud Computing (CLOUD), 2011 IEEE International Conference.*

[55] J. Z. Li, J. Chinneck, M. Woodside, and M. Litoiu, "Fast scalable optimization to configure service systems having cost and quality of service constraints," in *Proceedings of the 6th international conference on Autonomic computing*, 2009.

[56] A. Raveendran, T. Bicer, and G. Agrawal, "A framework for elastic execution of existing mpi programs," in *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*, 2011.

[57] J. Montes, A. Sánchez, B. Memishi, M. S. Pérez, and G. Antoniu, "GMonE: A complete approach to cloud monitoring," *Future Gener. Comput. Syst.*, vol. 29, no. 8, 2013.

[58] R. Chi, Z. Qian, and S. Lu, "A game theoretical method for auto-scaling of multi-tiers web applications in cloud," in *Proceedings of the 4 Asia-Pacific Symposium on Internetware*, 2012.

[59] P. Venkata Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2292–2303, 2013.

[60] F. Wuhib, R. Stadler, and M. Spreitzer, "A gossip protocol for dynamic resource management in large cloud environments," *Netw. Serv. Manag. IEEE Trans. On*, vol. 9, no. 2, 2012.

[61] J. T. Piao and J. Yan, "A network-aware virtual machine placement and migration approach in cloud computing," in *Grid and*

Cooperative Computing (GCC), 2010 9th International Conference on, 2010.

[62] Y. Jie, J. Qiu, and Y. Li, "A profile-based approach to just-in-time scalability for cloud applications," in *Cloud Computing, 2009. CLOUD'09. IEEE International Conference on*.

[63] M. Z. Hasan, E. Magana, A. Clemm, L. Tucker, and S. L. D. Gudreddi, "Integrated and autonomic cloud resource scaling," in *Network Operations and Management Symposium (NOMS), 2012 IEEE*, 2012.

[64] Y. Zhang, G. Huang, X. Liu, and H. Mei, "Integrating resource consumption and allocation for infrastructure resources on-demand," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 2010, pp. 75–82.

[65] R. Buyya, R. Ranjan, and R. N. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *Algorithms and architectures for parallel processing*, Springer, 2010, pp. 13–31.

[66] P. C. Brebner, "Is your cloud elastic enough?: performance modelling the elasticity of infrastructure as a service (iaas) cloud applications," in *Proceedings of the third joint WOSP/SIPEW international conference on Performance Engineering*, 2012, pp. 263–266.

[67] H. Kang, J. Koh, Y. Kim, and J. Hahm, "A SLA driven VM auto-scaling method in hybrid cloud environment," in *Network Operations and Management Symposium (APNOMS), 2013 15th Asia-Pacific*, 2013, pp. 1–6.

[68] W. Iqbal, M. N. Dailey, I. Ali, P. Janecek, and D. Carrera, "Adaptive resource allocation for Back-end Mashup applications on a heterogeneous private cloud," in *Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on*, 2010, pp. 317–321.

[69] M. Maurer, I. Brandic, and R. Sakellariou, "Adaptive resource configuration for Cloud infrastructure management," *Future Gener. Comput. Syst.*, vol. 29, no. 2, pp. 472–487, 2013.

[70] A. Kertesz, G. Kecskemeti, and I. Brandic, "An interoperable and self-adaptive approach for SLA-based service virtualization in heterogeneous cloud environments," *Future Gener. Comput. Syst.*, vol. 32, pp. 54–68, 2014.

[71] R. Buyya, R. N. Calheiros, and X. Li, "Autonomic cloud computing: Open challenges and architectural elements," in *Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on*, 2012, pp. 3–10.

[72] Y. W. Ahn and A. M. K. Cheng, "Autonomic computing architecture for real-time medical application running on virtual private cloud infrastructures," *ACM SIGBED Rev.*, vol. 10, no. 2, pp. 15–15, 2013.

[73] M. Mao and M. Humphrey, "Auto-scaling to minimize cost and meet application deadlines in cloud workflows," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, 2011.

[74] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, "Capacity management and demand prediction for next generation data centers," in *Web Services, 2007. ICWS 2007. IEEE International Conference on*, 2007, pp. 43–50.

[75] Z. Shen, S. Subbiah, X. Gu, and J. Wilkes, "Cloudscale: elastic resource scaling for multi-tenant cloud systems," in *Proceedings of the 2nd ACM Symposium on Cloud Computing*, 2011, p. 5.

[76] S. Abdelwahed, J. Bai, R. Su, and N. Kandasamy, "On the application of predictive control techniques for adaptive performance management of computing systems," *Netw. Serv. Manag. IEEE Trans. On*, vol. 6, no. 4, pp. 212–225, 2009.

[77] J. Povedano-Molina, J. M. Lopez-Vega, J. M. Lopez-Soler, A. Corradi, and L. Foschini, "DARGOS: A highly adaptable and scalable monitoring architecture for multi-tenant clouds," *Future Gener. Comput. Syst.*, vol. 29, no. 8, 2013.

[78] C. Vecchiola, R. N. Calheiros, D. Karunamoorthy, and R. Buyya, "Deadline-driven provisioning of resources for scientific applications in hybrid clouds with Aneka," *Future Gener. Comput. Syst.*, vol. 28, no. 1, pp. 58–65, 2012.

[79] S. Vijayakumar, Q. Zhu, and G. Agrawal, "Dynamic resource provisioning for data streaming applications in a cloud environment," in *Cloud Computing Technology*

and Science (CloudCom), 2010 IEEE Second International Conference on, 2010, pp. 441–448.

[80] J. Jiang, J. Lu, G. Zhang, and G. Long, "Optimal cloud resource auto-scaling for web applications," in *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, 2013, pp. 58–65.

[81] N. Janssens, X. An, K. Daenen, and C. Forlivesi, "Dynamic scaling of call-stateful SIP services in the cloud," in *NETWORKING 2012*, Springer, 2012, pp. 175–189.

[82] Y. Kouki and T. Ledoux, "SCAling: SLA-driven cloud auto-scaling," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 2013, pp. 411–414.

[83]W.-R. Lee, H.-Y. Teng, and R.-H. Hwang, "Optimization of cloud resource subscription policy," in *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, 2012, pp. 449–455.

[84] J. Nakajima, Q. Lin, S. Yang, M. Zhu, S. Gao, M. Xia, P. Yu, Y. Dong, Z. Qi, and K. Chen, "Optimizing virtual machines using hybrid virtualization," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, 2011, pp. 573–578.

[85] D. Shin and H. Akkan, "Domain-based virtualized resource management in cloud computing," in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010 6th International Conference on*, 2010.

[86] Z. Gong, X. Gu, and J. Wilkes, "Press: Predictive elastic resource scaling for cloud systems," in *Network and Service Management (CNSM), 2010 International Conference on*, 2010,

[87] G. A. Paleologo, "Price-at-Risk: A methodology for pricing utility computing services," *IBM Syst. J.*, vol. 43, no. 1, pp. 20–31, 2004.

[88] X. Wu, W. Zhang, and D. Wanchun, "Pricing as a Service: Personalized Pricing Strategy in Cloud Computing," in *Computer and Information Technology (CIT), 2012 IEEE 12th International Conference on*, 2012, pp. 1119–1124.

[89] Z. Liu, Q. Sun, S. Wang, H. Zou, and F. Yang, "Profit-driven Cloud Service Request Scheduling Under SLA Constraints," *J. Inf. Comput. Sci.*, vol. 9, no. 14, pp. 4065–4073, 2012.

[90] A. Corradi, M. Fanelli, and L. Foschini, "VM consolidation: A real case based on OpenStack Cloud," *Future Gener. Comput. Syst.*, vol. 32, 2014.

[91] A. Inomata, T. Morikawa, M. Ikebe, Y. Okamoto, S. Noguchi, K. Fujikawa, H. Sunahara, and M. Rahman, "Proposal and evaluation of a dynamic resource allocation method based on the load of VMs on IaaS," in *New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on*, 2011, pp. 1–6.

[92] R. N. Calheiros, R. Ranjan, and R. Buyya, "Virtual machine provisioning based on analytical performance and QoS in cloud computing environments," in *Parallel Processing (ICPP), 2011 International Conference on*, 2011, pp. 295–304.

[93] D. Niu, H. Xu, B. Li, and S. Zhao, "Quality-assured cloud bandwidth auto-scaling for video-on-demand applications," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 460–468.

[94] L. R. Moore, K. Bean, and T. Ellahi, "Transforming reactive auto-scaling into proactive auto-scaling," in *Proceedings of the 3rd International Workshop on Cloud Data and Platforms*, 2013.

[95] N. Ferry, A. Rossini, F. Chauvel, B. Morin, and A. Solberg, "Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems," in *CLOUD 2013: IEEE 6th International Conference on Cloud Computing*, 2013, pp. 887–894

[96] A. N. Toosi, R. N. Calheiros, R. K. Thulasiram, and R. Buyya, "Resource Provisioning Policies to Increase IaaS Provider's Profit in a Federated Cloud Environment," in *High Performance Computing and Communications (HPCC), IEEE 13th International Conference on* 2011.

[97] X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, "Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing," *Mob. Netw. Appl.*, vol. 16, no. 3, pp. 270–284, 2011.

[98] T. Knauth and C. Fetzer, "Scaling non-elastic applications using virtual machines," in

Cloud Computing (CLOUD), 2011 IEEE International Conference on, 2011, pp. 468–475.

[99] M. E. Frîncu, "Scheduling highly available applications on cloud environments," *Future Gener. Comput. Syst.*, vol. 32, pp. 138–153, 2014.

[100] C.-W. Huang, W.-H. Hu, C.-C. Shih, B.-T. Lin, and C.-W. Cheng, "The improvement of auto-scaling mechanism for distributed database-A case study for MongoDB," in *Network Operations and Management Symposium (APNOMS), 2013 15th Asia-Pacific*, 2013, pp. 1–3.

[101] W. Iqbal, M. Dailey, and D. Carrera, "SLA-driven adaptive resource management for web applications on a heterogeneous compute cloud," in *Cloud Computing*, Springer, 2009.

[102] M. R. Jam, L. M. Khanli, M. K. Akbari, E. Hormozi, and M. S. Javan, "Survey on improved Autoscaling in Hadoop into cloud environments," in *Information and Knowledge Technology (IKT), 2013 5th Conference on*, 2013, pp. 19–23.

[103] M. Ahmed, A. Chowdhury, M. Ahmed, M. M. H. Rafee, and others, "An advanced survey on cloud computing and state-of-the-art research issues," *Int J Comput Sci Issues IJCSI*, vol. 9, 2012.

[104] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on*, 2009, pp. 44–51.

[105] FlexiScale: http://www.flexiscale.com/.

[106] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in *Proceedings of the 16th ACM conference on Computer and communications security*, 2009, pp. 199–212.