1-1-2019

Virtual Environments Testing as a Cloud Service: A Methodology for Protecting and Securing Virtual Infrastructures

Elhadj Benkhelifa
Staffordshire University

Anoud Bani Hani
Zayed University

Thomas Welsh
Staffordshire University

Siyakha Mthunzi
Staffordshire University

Chirine Ghedira Guegan
Université Jean Moulin Lyon 3

IEEE Access Multidisciplinary : Rapid Review : Open Access Journal

SPECIAL SECTION ON ADVANCED SOFTWARE AND DATA ENGINEERING FOR SECURE SOCIETIES

Received February 28, 2019, accepted March 15, 2019, date of publication April 23, 2019, date of current version August 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2912957

Virtual Environments Testing as a Cloud Service: A Methodology for Protecting and Securing Virtual Infrastructures

ELHADJ BENKHELIFA[1], (Member, IEEE), ANOUD BANI HANI[2], THOMAS WELSH[1], (Member, IEEE), SIYAKHA MTHUNZI[1], AND CHIRINE GHEDIRA GUEGAN[3]

[1]Cloud Computing and Applications Research Laboratory, Staffordshire University, Stoke on Trent, UK ST4 2DE, England
[2]College of Technological Innovation, Zayed University, Dubai 19282, United Arab Emirates
[3]IAE Lyon School of Management, University Lyon 3, 69008 Lyon, France

Corresponding author: Elhadj Benkhelifa (e.benkhelifa@staffs.ac.uk)

ABSTRACT Testing is a vital component of the system development life cycle. As information systems infrastructure move from native computing to cloud-based and virtualized platforms, it becomes necessary to evaluate their effectiveness to ensure completion of organizational goals. However, the complexity and scale of virtualized environments make this process difficult. Additionally, inherited and novel issues further complicate this process, while relatively high costs can be constraining. Enabling service-driven environments to provide this evaluation is therefore beneficial for both providers and users. No such complete service offering currently exists. This paper is therefore aimed to benefit industry and academia involved in areas involved with cloud-based testing of virtualized software and its environments. A review of current literature highlights a number of challenges in the domain. An analysis of the challenges aided in deriving requirements for developing a servitization framework for virtual infrastructure testing as a service. It is anticipated that this framework can further feedback into developing solutions to the aforementioned challenges. An evaluation of a real-world organization's servitization requirements case scenario indicates that the proposed framework provides potential solutions for associated use cases.

INDEX TERMS Testing, servitization, virtual environment, utility computing, cloud computing, security, testing as service.

I. INTRODUCTION

Virtualization technologies are foundational components for a wide variety of computing scenarios. Initially employed in numerous roles for enhancement of operating systems, virtualisation has now grown to enable full representation of computing hardware at near native speed [1]. The aggregation of differing forms of virtualisation has allowed the construction of complete information system environments and through which has enabled the paradigm of cloud computing. Whether cloud-based or not, these virtual system environments afford a variety of benefits over information systems constructed directly on native hardware. As well as ease of general system construction and management, economy is often seen as a main driving factor for adoption of virtualised environments, often at the expense of mild performance degradation [1].

As with their traditional counterparts, it is essential that these systems operate within their given constraints,in line with the goals of their governing organisation. Whilst the criteria to be assessed will typically remain the same, the methods for assessing vary. In some ways, it has become easier due to the added capabilities of automating previously physical tasks. Nonetheless, the complexity and cost of the environments generates new challenges for the evaluation of these virtual ecosystems [2].

In addition to the aforementioned factors, another driving factor for this work is from the migration to cloud-computing, which has given way to a plethora of computing-based services. With its foundation of virtualisation

The associate editor coordinating the review of this manuscript and approving it for publication was Mahmoud Barhamgi.

technologies, it seems a suitable choice to provide evaluation services for virtualised environments. However research has revealed that there is no complete offering in this area, with complexity of the environments and testing process appearing to be a dividing problem [3]. Therefore to further enable the development of cloud-based virtual infrastructure testing, this paper provides a survey of works in this area, foremost to further understand the challenges associated with the development of cloud-based testing environments. In order to drive development in this area, a framework for servitisation of testing within cloud-based environments is derived and presented.

The focus of this paper can therefore be considered as an intersection between the following. The increased uptake of virtual environments has generated a novel necessity for their performance evaluation, with the addition of the move to utility based cloud-computing, which has servitised most aspects of information system offerings; ensures the integration of virtual environment evaluation and cloud-based services is inevitable. Such a cloud-based virtual infrastructure testing service is beneficial not just as a lucrative offering for cloud providers, but also to recursively ensure that cloud environments are operating effectively and efficiently.

As one primary factor for driving cloud adoption is to increase economy, ensuring then that the cloud service is economical for all involved actors is an undeniable priority. When performing a survey of the aforementioned similar and relevant areas: cloud based-servitisation, cloud-based testing and virtual environment evaluation, a number of issues become apparent. Therefore, this paper focuses upon the challenges seen at this intersection, the analysis of which then helps to drive forward virtual infrastructure testing as a service through a thorough derivation of challenges and their inter dependencies through which has facilitated the development of a framework for a more complete integration of software testing, cloud-based testing, virtual environment evaluation and servitisation areas.

A. METHODOLOGY

Cloud-based inventions and adoptions are a growing trend since the advent of the cloud. Traditional software testing can arguably innovate to use cloud for testing various environments.

In this survey paper, we begin by evaluating and interpreting outcomes from a range of works in software testing. Through this meta-synthesis, we derive logical conclusions, and present them in brief prefaces to the entirety of the current research. Having said that, we formulate three chronological questions whose purpose are to, first, guide the reader through the discussions in the current section, and in addition, to enhance and conceptualise the above-mentioned meta-synthesis. We thus pose the following questions:

1) What are we testing?
2) How are we testing it?
3) Where do we conduct the testing?

Through a synthesis of the aforementioned, we can identify the common and core elements for testing, and transform our findings into new conceptualisations. Further to the foregoing discussion, we categorically state the need for testing virtual environments and provide a comparative analysis of testing considerations thereof. Data yielded from the aforementioned analysis fosters discussions in proceeding sections.

Our observation is that, many of the challenges in current cloud-based testing solutions from both industry and academia, are a result of the cloud environment, the complex nature of the environments to be tested, and the testing process itself. Furthermore, we observe that cloud-based virtual infrastructure testing as a service is a niche market from both an academia and industry perspective. As such, to the best of our knowledge, the collective entirety of this paper is the first of its form.

In order to mitigate the challenges mentioned above and successfully drive solutions for cloud-based virtual infrastructure testing forward, we propose a framework for cloud-based virtual infrastructure testing servitization. To accomplish this, we first derive requirements for a service through an analysis of the challenges noted in the previous section. Next, we derive and framework servitization process requirements from other industries. We finally map the requirements to the aforementioned process, in line with commonly used guidelines for producing a service.

II. BACKGROUND

In engineering systems development, testing is a vital process hugely linked to the cost of developing and maintaining software or infrastructure [4]. It is primarily aimed to raise confidence about a system by measuring if the behaviours of the intended and actual system differ [5]. In Model-based testing for instance, testing is guided by unambiguous behaviour defining models that mimic validated, reproducible, and documented behaviours of a system and its environment [5]. In the computing continuum specialising in information systems delivery, it is incumbent upon the organisation to ensure that its deliverables, including the underlying infrastructure are robust and efficient enough for reliable delivery of services. As such, it is important to understand testing techniques and assess challenges and opportunities thereof, so as to predict their performance in varying environments [4]. Thus, an in-depth understanding of the challenges and opportunities in fault-revealing test methods, of associated costs, and testing cost-effectiveness, enhance the development of a robust test hypothesis, including the necessary test controls [4]. Thus conceived, we formulate the following three chronological questions as guides for discussions in the later sections; what are we testing? How are we testing it? Where do we conduct the tests? We postulate that in addressing the above questions, considerations for testing such as adequate planning can be optimised.

TABLE 1. Advantages of cloud-based testing.

Cloud-based testing	Benefits of Cloud-based testing
Leverages Cloud computing infrastructure for testing environments and activities	Ability to create multiple scalable, on-demand cloud environments and test configurations to accelerate development and testing cycles
Ability to simulate web-traffic and production style network state at minimal costs for testing purposes is a strong selling point	Ability to closely replicate live production environments for better and more accurate testing process
Entails quick and easy auto-provisioning of computing resources and infrastructure which can include virtual resources	Increase test coverage, test cycles and increase bug resolution cycles
On-demand test execution of test cases	More flexibility/agility to meet varying business testing demands
Testing can be done via the internet connection as a service at any time amongst a geographically dispersed team	Enable easy collaboration for multiple product and release teams, including secure, limited access for contract or off-shore test teams. Share bug snapshots with remote development and testing teams.
Can be less expensive for large-scale software testing projects over any length of time due to CloudâĂŹs pay-per-use pricing	Create isolated test environments for broad range of OS, database, browser, and application
Ease of scaling, saving or recreating test environments for faster defect resolution and faster time to market for products	Easily recreate use cases. Enable rapid defect resolution by capturing the entire state (memory, network settings, and disk) of a multi-machine configuration and saving as templates

A. WHAT WE TEST - INFRASTRUCTURE TESTING

Testing criteria depend on what test elements to pay attention to, the test scenarios, and test solutions used. A line of argument suggests the notion that in communication infrastructure testing such as session initiation protocol (SIP), performance testing is underdeveloped with no unanimous recommendations on how to perform the test [6]. In this respect, the authors posit the argument that existing proprietary solutions provide comprehensive test scenarios, with inhibitive costs nonetheless, often providing incompatible test results due of their proprietary nature [6].

Software testing: Software testing is viewed synonymous with validation testing, primarily conducted to discover errors that exist in software as a result of inadvertent poor design and construction [7]. We refer the reader to [7] for detailed information. In distributed software environments with integrated platforms, the need for reliable, scalable and fast service delivery makes it is not uncommon that performance testing should include the evaluation of latency, scalability, and throughput [8]. As an online service, software testing is argued to provide web-based route maintenance, daily operation and testing support via frameworks and servers [9].

B. HOW WE TEST - FUNCTIONAL TESTING

In the software testing continuum, functional testing describes the criteria for verifying that the behaviour of a software component complies with corresponding specifications [10]. In so doing, functional testing does not take into account how long a process takes, when results from a process are produced, or how much resources a process requires [10] or the details of the software in order to evaluate its behaviour [11]. The foregoing imply that functional testing alone cannot be relied upon to provide a totally acceptable system. On the other hand, software testing is a costly process, and most suitable for large businesses. In contrast however, due to the costs involved in testing infrastructures, a majority of small enterprises only perform functional testing [12]. Typically, some of the functional requirements in functional testing include:

- Administrative functions and Business Rules
- Authentication and Authorisation level
- Audits and External interfaces
- Legal and Regulatory requirements

Non-functional testing: In contrast to testing to establish behavioural changes in modified software [13], non-functional (white-box) testing evaluates the operation of a system rather than its behaviour, making it pivotal in the successful completion of all nontrivial software [14]. This argument lies in the notion that traceability, an attribute of non-functional testing, is indirectly linked to activities that would traditionally be classified as functional [14]. Table 1 below summarises the attributes of functional and non-functional testing and their corresponding testing types. Depending on the test scenario, the following requirements are generally associated with non-functional testing, in practice however, this list may vary; Performance, Security, Scalability, Survivability, Availability, and Interoperability.

C. WHERE WE CONDUCT TESTS - NON-CLOUD-BASED

Two main testing processes in the traditional context are; manual testing, which is testing without a tool or script to identify unexpected behaviour; and, automation testing which uses test or other software in testing. Automation is often idea in testing types such as regression testing. However, this process is not without its challenges, for instance, difficulties with performing exhaustive testing, the costs of infrastructure and limitations of time when testing large projects. Furthermore, setting-up traditional test environments can be time-consuming, while some testing activities can be slow, thereby reducing TTM e.g. regression testing. Additionally, low rate of rapid defect resolution due to difficulty in capturing state (memory, network settings

TABLE 2. Virtualisation measurement levels.

Layer	Description	Focus
Single Server	Measurement of one or more VMs on a single server	CPU, Memory, Network bandwidth, Disk bandwidth
Single Data Centre	Measure of VM interaction across multiple servers	Live Migration, Deployment, Snapshotting, Shared Storage, Shared Memory
Multi Data Centre	Measurement of VM interactions across geo-distributed data-centres	Live WAN VM Migration, Live WAN Storage Migration

and disk) of multi machine configurations, and difficulties in enabling collaboration for multiple product and release teams introduce challenges.

D. CLOUD-BASED TESTING

As reported by IBM, Cloud testing could potentially realise a reduction of 50%-70% in Licensing and capital expenses due to the use of virtualised resources; and a reduction in labour and operating cost of 30%-50% due to automated testing and resource provisioning. It could also improve overall product quality & reduce the need to detect defects by 15% - 30%. A line of argument suggests the notion that with the advent of cloud computing, telecommunications and web-based applications are particularly primed for online testing [9]. This is grounded in the logic that with the cloud being an online service, online applications will be better tested in their host environments [9]. Cloud computing is increasingly changing the way software products and services are produced and consumed, thereby implying the need for a change in the ways, methods, tools and concepts by which these are tested. This is important considering that Global IT software spending was estimated at $267bn in 2011 and 20-30% of SDLC time is spent on testing. The Global testing market is worth $46bn, while testing software tools alone are worth $2bn (Gartner).The advent of cloud computing has brought new possibilities in the way testing infrastructures can be provisioned, setup, accessed, utilised and managed. At the same time, cloud computing has also brought to limelight some issues in traditional testing that need to be addressed. The three most common perspectives of cloud-based testing are, testing applications for the cloud, testing applications and/or environments in the cloud, and testing the cloud. Although Cloud-based technologies present a number of advantages and benefits over traditional testing (Table 2), it cannot overly replace traditional testing; as areas and scenarios of testing for synergy and trade-offs exist [1]. For example, some testing areas requiring implicit domain knowledge about the customer's business (like insurance business); or areas where hardware or software is an integral and essential part of the other and directly dependent on each other (like programmable logic controllers), may require the adoption of traditional testing practices over cloud-based testing [2]. The following are some of cloud computing's key benefits, and further considerations for cloud-based testing:

- Scalability: Cloud computing delivers seemingly infinite computing capabilities over an otherwise finite underlying infrastructure [15], meaning that resources are accessible on-demand according to requirements. In addition, spent on testing; that is 20-25% of the total IT spend.
- Cost Effectiveness: Cloud is lauded for keeping overheads at a minimum [16] by enabling requirement customisation, as opposed to acquiring expensive systems, with incomparable scalability [15].
- Accessibility: Cloud computing facilitates access to a wide variety of tools, applications, and web services in a user-centric manner [15].

III. VIRTUAL ENVIRONMENTS (VE)

Vitalisation describes the concept of creating an abstract representation of something, as evidenced in a wide range of systems; visualised biological structures in medicine and health to architectures in engineering, and design and analysis of manufacturing systems to military training [17], whereupon the virtualization concept possesses stark similarities [1]. Virtuality in computing is multi-dimensional, on one end it is viewed as reality (e.g. as viewed by programs and applications), while on the other, it is represented differently in its underlying formal structures [1]. Arguably, it is these characteristics that necessitates the argument that virtualization is indeed a revolutionary development for computing [18]. Parmelee et al describes virtualization as "a distinction between the logical address and physical address" [19]. In a general sense, virtualization itself has historically been seen throughout computer science, for instance with virtual memory [20], virtual networking [21] and virtual storage [19] (arguably the birth of conventional file systems). Indeed, since its development by IBM Corporation in the 1960s [1], there has always been a need to develop abstractions of hardware in software, to bypass hardware limitations and enable seemingly unlimited resources. However, it could be argued that current virtualization technology is the aggregation of all these concepts into a single package, and therefore is more of an inevitability of the continued requirement to optimise hardware usage, as opposed to a revolutionary concept.

Therefore, throughout this paper we refer to virtualization environments as those conducted by virtual machines (VMs) and hypervisors, and not virtualization in the broader view. The exception to this rule is Linux containers, or Operating System (OS) level virtualization, which will be discussed later. In the following subsection, we provide a brief classification of virtualization approaches according to their underlying isolation technologies, and briefly outlines each approach's advantages and disadvantages.

A. BENEFITS OF VIRTUALISATION

Vitalisation allows a variety of benefits such as pausing, cloning and migrating machines from one hypervisors to another. Virtualized network management facilitates a variety of easy to manage network scenarios e.g. isolated networking and network production with no physical network hardware. Aggregation of hardware and hypervisors also allows arbitrary resource allocation, giving virtual machines performance which would be otherwise unobtainable. Likewise, the ease of management of virtualized resources enables resources to be provisioned and scaled dynamically, allowing more hardware to be allocated as needed. This is a primary feature of cloud computing environments, and is therefore illustrative of the massive benefits that virtualization technologies provide. Furthermore the ability to consolidate small clusters of servers into a single more powerful virtual server not only reduces hardware costs, but means that legacy applications can be migrated with relative ease [1]. However, the above mentioned virtualization techniques inevitably incur resource overhead. It has been shown that performance can vary considerably depending on the application employed [22], such as when utilising virtual clusters [23] or for specific resource requirements such as I/O intensive applications [23]. The distinction between evaluating the performance of virtual infrastructures and conventional ones is purely because conventional system benchmarking fails to take into account of additional features which virtualization enables(e.g. resource sharing).

B. NEED FOR TESTING VE

An increasing number of organisations are taking advantage of virtualized environments to involve their stakeholders and clients in product development and support activities [24]. However, the complexity of VE, and modern VEs in particular, presents major challenges to testing activities. Modern VE need to be extensively tested before being implemented, and before changes can be rolled out into live environments. In spite of the foregoing, It is expensive & impracticable for most customers to maintain sufficient resources to carry out proper testing. Nevertheless, with an increased availability of computational power and innovative technologies, VE testing just like software testing, aims to evaluate the quality of a VE environment [25]. Due to the variety of virtualization methods, it is necessary to evaluate the reliability, availability, consistency, transparency, and performance of VE. VE testing is particularly pertinent in a scenario where a VE hosts a high performance computing application, to ascertain if a VE can meet the high performance requirements of a (HPC) [16].

NIST estimates that software errors account for $59.5B of losses each year in the US alone. While it seems attractive for users to build their testing infrastructures in the Cloud using Amazon or Rackspace, current testing methods tend to involve building the testing infrastructure in old hardware on-premise. Neither Amazon nor Rackspace provides off-the-shelf capability of testing VE. For SMEs and indeed larger organisations, using Amazon services or on-premises, both prove cumbersome due to associated costs, resources & required expertise & administrative overhead. In both options, users will need to procure, install and configure complex testing tools. Invariably, they may fail to meet their objectives because of (a) insufficient resources to create and maintain a test infrastructure regardless of the platform being used (b) the cost, complexity & inflexibility of available testing.

Leveraging Cloud computing capabilities could provide the opportunity to ensure that software quality assurance is continuously delivered to customers in a cost-effective manner in line with the evolving technology. This can be accomplished by addressing the shortcomings of current traditional software testing practices. A variety of attributes considered for evaluation when testing VEs are contingent upon the use case, and the structure of the virtualized environment; a single server, a single data centre, or multiple data centres. Considerations about which features are measured vary from one scenario to the other, for instance, test case pertaining to VM interactions localised to a single server, or VM interactions on multiple servers but localised to a single data centre, or VM interactions on servers in multiple data centres, etc. (See summary in the table 3 below).

TABLE 3. Factors affecting virtualisation performance.

Type	Description
VM Size	The initial size of the VM resource allocation (memory, CPU, storage)
Co-located VMs	The number and type of applications located on the same HV
Hypervisor Technology	Specific software used and its support
VM Consolidation	Aggregation of multiple VMs into a single VM
VM Scaling	Scaling the quantity or size of a VM
VCPU Pinning	Allocation of a VM to a specific PCPU
VMExits	HV switching focus to another VM

Additionally, further considerations are particularly imperative when conduction testing in, or when testing distributed environments. For instance, the analysis of the testing process is an important and fundamental technical procedure, which helps derive what to test and how to test it. This process focuses on measurements of resource on the local machines, and how these resources can be measured across multiple layers. Furthermore, interference factors with collocated VMs in virtualized environments pose a significant challenge of cross-talk between collocated VMs. Optimising performance evaluation for testing ensures a clean and consistent testing procedure and environment,subsequently, it is expected to reduce the number of tests while optimising resources.

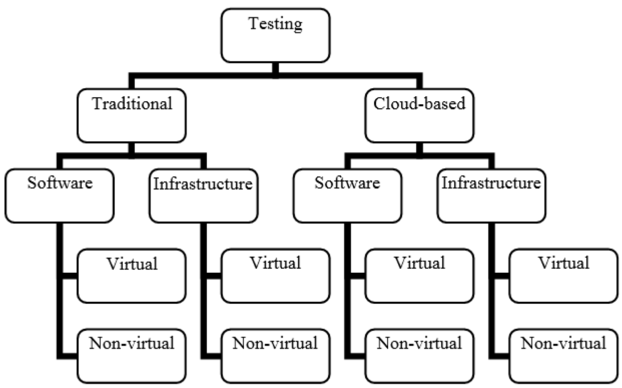

FIGURE 1. Testing categories.

The table below summarises factors which affect VM performance.

Our study corroborates current literature on the challenges with VE testing, but extended research to testing complex virtual environments. We suggest the following as issues specifically affecting cloud environment testing. Foremost, as clouds are typically geo-distributed, issues relating to measuring VM migration including monitoring are complicated by the lack of information on when and where a VM might move to. Additionally, particular consideration ought to be paid for resource restrictions, as hardware and costs are an important consideration in testing cloud environments. This is based on the premise that, since cloud environments seemingly exhibit unlimited resources, it becomes a challenge to evaluate other cloud platforms without the need for even higher resources. This cost benefit consideration is further compounded by the fact that, cloud environments are built on a pay-as-you-go model, meaning that where testing is uncertain, it is a challenge to predict and plan for its costs.

Moreover, inaccuracies in measurement are a likely occurrence due to the dynamic and complex nature of the cloud.

IV. STATE OF THE ART: CLOUD BASED TESTING

From the above cloud related testing can be categorised in three perspectives as described below. Figure 2 classifies testing in terms of where to test and what to test as discussed above.

- **Testing virtualised environments** Reviewing the work within this area was important in order to understand the processes that were used to evaluate and test virtual environments.
- **The cloud as a testing environment** Whilst this work does not focus on testing virtualized environments, it provides an insight into the fundamentals of cloud management when leveraged for software testing.
- **Testing cloud environments** Cloud environments are inherently complex due to their large scale, distributed nature and dynamic structure. Literature in this area elaborated on the challenges of testing complex virtualized environments.

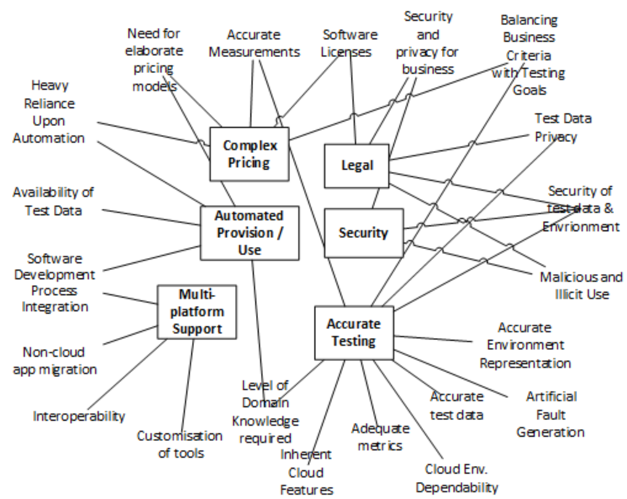

FIGURE 2. Requirements to challenges mapping.

This section details and analyses a review of academic research and industrial development and practice in cloud testing. Including related literature is an essential component of the research, primarily in order understand the current state of the art in various systems. As opposed to producing a comparison between the systems and research, a summarised description of the work will be given in Table below, to provide an overview of current literature. The results of an examination into prior research attempts, will facilitate greater insight into any gaps area, and provide basis for future research and development efforts.

A. CLOUD-BASED TESTING IN ACADEMIA

A line of argument in literature suggests the notion that, with the advent of cloud computing, telecommunications and web-based applications are particularly primed for online testing [9]. This notion is grounded in the logic that, with Software as a Service, online applications will better be tested in their host environments [9]. When reviewing the body of work which covers cloud-based virtual software testing, there are a number of sub-fields which may be considered as each one contributes to the general area. This body of work includes: software test suites which leverage the cloud, testing of cloud applications at each layer of the cloud (Infrastructure, Platform and Software), testing of virtualized environments (in a non-cloud setting), software which provides testing as a cloud service and a general evaluation of experiences used in testing virtualized environments, including further research directions.

When exploring the uses of the cloud for software testing, In [26], Koa et al present a framework which leverages the features of the cloud to enable a full software development and testing environment, although they present only a high-level overview and no implementation or analysis of the system. The framework does, however, break the testing components into individual components which are bespoke.

TABLE 4. Literature in virtual infrastructure testing as a service.

Authors	Description	Type
Kao et al 2014 [26]	Design of a component based system use for testing software in cloud environments. Example: Web Application	Cloud-based software testing
Riungu et al 2010 [9]	Qualitative study where managers were interviewed on their opinions of software	
Hanawa et al 2010 [27]	Leverages full cloud features using eucalyptus to perform automated testing with an emphasis on fault injection in distributed software environment.	
Robinson and Ragusa 2011 [3]	A method referred to as âĂIJtesting the cloudâĂİ from an infrastructure point of view.	
Riungu et al 2010 [2]	Provides a brief overview of testing software within cloud environments via interviewees from eleven different organisations.	
Ciortea 2010 [28]	STaaS utilising symbolic execution for increased performance.	
King and Ganti 2010 [29]	Enabling cloud software environments to perform ATS in tandem with Test-support-as-a-Service to provide developers with comprehensive.	Non cloud-based testing of cloud environments
Citron and Zlotnick 2011 [30]	A hybrid simulation and emulation method for testing large cloud environments.	
Gao et al. 2011 [31]	Formal models and approaches to evaluating SaaS performance and scalability.	
Shi et al 2015 [32]	CloudTB: A quick and reliable testbed for virtual machine based cloud computing systems.	
Lynch, Cerqueus and Thorpe 2013 [33]	Evaluates novel tools and methodologies developed for testing IBM SaaS applications.	
Tomasson 2013 [34]	Evaluated testing a cloud application on the amazon ec2 platform.	
Lim et al 2013 [35]	Assessing the performance of a data analytics program during resource changes *sizing*	Non-cloud based performance testing of virtual environments
	Provides methods for quantification of disk I/O as a metric for VM performance	
Mas'ud, Yaacob, and Ahmad 2006 [36]	Network I/O evaluation of a VM based IDS.	
Al Jabry, Liu, Zhu, and Panneerselvam, 2014 [37]	A Type 2 hypervisor performance comparison.	
Martinovic, Balen, and S Rimac-Drlje, 2010 [38]	A performance comparison when adjusting the underlying OS.	
Chi, Qian and Lu 2014 [39]	Numerous VM collocation performance degredation effects.	
Zhu, Zhu and Agrawal 2010 [40]	KCCA VM performance model for power optimisation	
Koh et al 2007 [41]	Application type performance analysis	
Hashimoto and Aida 2012 [42]	Collocation performance degradation within HPC applications	
Pu et al 2010 [43]	Shows degredation of performance during Collocation due to disk and network I/O	
Tickoo et al 2010 [44]	Describes numerous challenges to providing accurate performance to virtual machines.	
Kundu et al 2012 [45]	Machine learning performance models.	
Soundararajan et al 2014 [46]	Multi-layer VM Benchmarking	
Ye et al 2014 2014 [22]	A full featured virtualisation benchmarking suite.	

This allows greater integration with cloud environments and thus further enables and catalyses the features such as visualisation and scalability due to the modular fashion.

Another cloud system for software testing is presented by Hana et al [27]. This system is focused upon testing large-scale dependable and parallel systems, with the authors arguing that as systems become larger, their complexity increases which in turn creates further difficulties in testing. They propose to mitigate these issues through leveraging the features of the cloud. The authors explain that dependable systems rely on a great degree of redundancy in order to be reliably dependable and that the additional components cause the system to become more complex which creates issues in tracing faults when they occur; a factor which heavily influences software testing. The proposed solution involves the development of a VM with specialised fault injection and tracking features enabled through the eucalyptus cloud software.

A test bed called Cloud9 is presented in [28]. Marketed as a software testing service and not specifically for virtualized environments, the work highlights the efficiency of parallel execution built upon the cloud; utilising symbolic execution to increase performance.

The authors in [9] provide some valuable insight into cloud-based software testing. They present a study which used a grounded theory approach to understand more about the software testing as a service field, in order to provide valuable insights into the software testing as a service domain.

TABLE 5. **Market analysis of Key commercial players in testing for virtual and non-virtual environments.**

V:Virtual Environments; NV: Non-Virtual Environments; C: Cloud-based; NC: non-Cloud-based; X: Cross-platform

Testing components	UTest		Sauce Labs		Skytap		HP		IBM		SOASTA		Fujitsu		Microsoft		Sogetti		Login VSI		Oracle		VMware	
Non-Functional Testing	V	NV	V	NV	V	NV	V	NV	V	NV	V	NV	V	NV	V	NV	V	NV	V	NV	V	NV	V	NV
Security Testing		C				C-X		C-X		C-X		C-X			NC	C-X		C	NC				NC	
Performance Testing		C		C-X		C-X	NC	C-X	NC	C-X		C-X	NC	NC	NC	C-X		C	NC		NC	C-X	NC	
Load Testing		C					NC	C-X	NC	C-X		C-X	NC		NC	C-X			NC		NC	C-X	NC	
Usability Testing		C					NC	C-X	NC				NC		NC	C-X		C	NC		NC	C-X	NC	
Compatibility Testing		C		C-X		C-X		C-X	NC			C-X	NC		NC	C-X		C			NC	C-X	NC	
Compliance Testing								C-X		C-X												C-X		
Functionality Testing		C		C-X		C-X	NC	C-X	NC	C-X		C-X	NC		NC	C-X		C	NC			C-X	NC	
Interoperability Testing		C		C-X				C-X								C-X		C						
Localization Testing		C				C-X																		
Maintainability Testing				C-X																		C-X		
Stress Testing		C				C-X		C-X	NC				NC		NC	C-X			NC			C-X	NC	
Scalability Testing				C-X		C-X		C-X				C-X	NC		NC	C-X			NC		NC	C-X		
Recovery Testing								C-X																
Test automation				C-X		C-X		C-X	NC	C-X		C-X	NC	NC		C-X		C	NC		NC	C-X		
Test management				C-X	NC	C-X		C-X	NC	C-X						C-X		C	NC			C-X		
Functional Testing	V	NV	V	NV	V	NV	V	NV	V	NV	V	NV	V	NV	V	NV	V	NV	V	NV	V	NV	V	NV
Unit Testing		C		C-X		C-X		C-X		C-X		C-X		NC		C-X		C				C-X		
Smoke/Sanity Testing		C		C-X		C-X												C						
Integration Testing		C		C-X		C-X		C-X		C-X		C-X		NC		C-X		C				C-X		
Interface/Usability Testing		C		C-X		C-X		C-X		C-X		C-X	NC	NC		C-X		C	NC			C-X	NC	
System Testing		C		C-X		C-X		C-X		C-X		C-X				C-X		C						
Regression Testing		C		C-X		C-X		C-X		C-X		C-X				C-X		C				C-X		
UAT		C		C-X		C-X		C-X		C-X		C-X		NC		C-X		C						

A number of research issues were identified from their research. Domain knowledge was cited as a problem by some respondents for example with embedded systems, where using the cloud to test these systems was not particularly feasible. Other issues were those such as security, which is found throughout cloud-based services.

Further research issues directed at software testing in cloud computing are presented in [2]. As with the previous study, these issues are gathered from interviewing various actors within the software testing process, such as managers, CTOs VPs etc. The issues were grouped into application based (considered as technical issues), business and legal. The application based issues relate to ensuring an application is suited to cloud testing, available for a wide number of use cases, guaranteeing the reliability of the testing process and interoperability. Management issues include ensuring an availability of testers and the impact upon the business whilst legal relate to security of the test data and the accuracy of SLAs. The authors stress the need for pilot studies for cloud based software testing in order to understand more about the numerous issues.

A summary of the academic literature which was surveyed is presented in table 5, which is organised according to the three categories of cloud related testing, stated in Section 4; to recall, these are (a)Testing virtualised environments; (b)The cloud as a testing environment and (c)Testing cloud environments. An analysis of the literature shows minimal work in all of the above categories. This is a logical fining, since cloud computing has only emerged during the past ten years. However, out of these testing categories, testing Virtual Environments seems to be the one area, which has received less attention, although, we acknowledged in Table 5, some related contributions, but almost all of which cover the area of performance related evaluation of different virtualisation configurations and performance modelling and prediction.

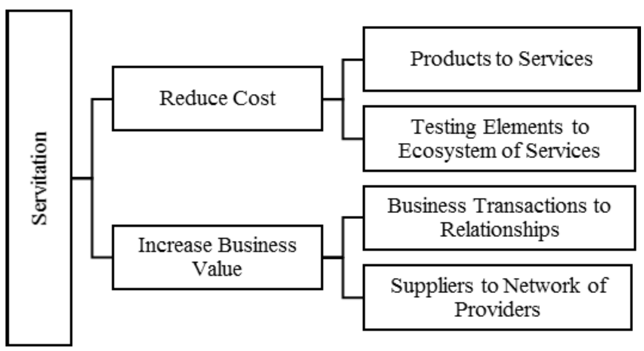

FIGURE 3. **Cloud-based software testing servitisation framework.**

B. CLOUD-BASED TESTING IN INDUSTRY

This section reviews cloud-based testing within industry. These tools include: self-learning test case libraries and tools for measuring or benchmarking cloud-based testing services for virtual and non-virtual environments. Figure 3 provides a summary of an assessment of market based literature.

A recent study of Market Research Media forecasts that U.S. government spending on cloud computing is entering an explosive growth phase in the coming years, with expenditure surpassing seven billion dollars by 2015 [3]. Logically, this forecast has a direct effect on related cloud-based services. The success of Amazon, eBay and Google has led to the rise of cloud computing as a new, proven architecture of how the traditional datacentre is built and managed. Indeed, recent publications from various respectable organisations provide encouraging forecasts, and to a greater extent, reason for optimism. For instance, International Data Corporation's (IDC) suggestion that worldwide spending on public cloud services reached $47.4 billion in 2013, increasing to over $107 billion in 2017 [6]. Similarly, predictions by Gartner estimating the cloud service market would reach $150.1 billion in 2013 [4] insinuates that there is vast scope for businesses and future developments in the cloud service market.

Recent years have witnessed an increase in the demand for thin client technology [20]–[22]. Thin Client computing is a way of maintaining computational services at a reduced total cost of ownership (TCO). Along similar lines, there are efforts from academia as well as industry's key players towards research and development of cloud-based testing solutions for applications, web sites, and other services. However, there are no fully developed cloud-based service offerings for testing virtual environments [1], [12]. As illustrated in Table and Table, the closest competitors in this area are Fujitsu, VMware and IBM who offer very limited and on-premise solutions for testing, but so far none of them offer a Cloud-based infrastructure for testing VE. Similarly, there are other solutions such as LoginVSI and SwiftTest for testing Virtual Infrastructures, but they are not Cloud-integrated & are also highly costly, complex and cumbersome to use.

The complexity of modern VE presents major challenges to testing activities. These systems need to be extensively tested before being implemented and before changes can be rolled out to live environments. It is expensive and impracticable for most customers to maintain sufficient resources to carry out proper testing. While it seems attractive for users to build their testing infrastructures in the Cloud using Amazon or Rackspace, current testing methods tend to involve building the testing infrastructure in old hardware on-premise. Neither Amazon nor Rackspace provides off-the-shelf capability of testing VE. Also, for SMEs and even for larger organisations, using Amazon services or on-premise, both prove cumbersome, due to associated costs, resources & required expertise & administrative overhead. In both options, users will need to procure, install and configure complex testing tools. Invariably they fail to meet their objectives because of:

1) insufficient resources to create and maintain a test infra-structure regardless of the platform being used
2) the cost, complexity and inflexibility of available testing tools
3) inflexibility of the Cloud Provider to meet customers' changing requirements.

Google trends show that there had been rising interest in VDI, but there has recently been a drop in interest. some views and opinions have attributed this to the the lack of adequate testing and support tools to enable accurate planning and deployment of VDI environments, more specifically, that meet customers' requirements.

We put forward the claim that, servitization of cloud-based virtual infrastructure testing will ensure that software quality assurance is continuously delivered to customers in a cost-effective manner.

While the idea behind servitization is to devise a system that fits a competitive business model based on value, and provides propositions for customers, based on benefits and tread-off, [17] it also offers an opportunity for customers to fulfil aspects of the service [17]. A common example of servitized services in Rolls Royce, whose business value comes from, and whose responsibility lies in servicing and maintaining engines rather than the sale of engines (Black Pepper, 2012). The concept of servitization rest on five primary complimentary considerations; a shift from products to solutions, outputs to outcomes, transactions to relationships, suppliers to network partners, and elements to ecosystems of complex products. To a large extend, servitization exists in IT industry particularly with the advent of the cloud, where software is offered and delivered as a service, i.e. SaaS. In the greater scheme of things, the offering and delivery of services in cloud computing removes the burden of upfront hardware and software costs, including their maintenance and arguably generates huge revenues [47].

Service-oriented processes such as service-oriented architectures, service-oriented computing, and service-oriented infrastructures, are a growing trend in both academia and industry, as a means for achieving greater business integration [17]. Just as cloud computing augments traditional information technology; hardware, software, networking, and infrastructures, etc. to provide computing as a service, service-oriented testing and service-oriented VE testing in particular, can evolve to mitigate some of the current business and technical challenges. Foremost, service-oriented functional requirement gathering from users can be enabled through the web in a SaaS-centric Web App using APIs.

However, despite the numerous features of the cloud which inherently enable servitization of a wide variety of use cases, the aforementioned gap within academic literature and marketresearch which surrounds the lack of offerings in the area of virtual infrastructure testing as a service, cannot be ignored.

An analysis of this literature highlights a number of key elements. Specifically, whilst the myriad of previous work available shows that the technical capabilities of performing such testing is possible, the primary clash between cloud platforms and the servitisation of virtual infrastructure testing can be summarized as the need to cover a wide variety of heterogeneous software and hardware environments, having a huge detrimental impact upon the economy of offering the service within the cloud. Every time an additional platform, architecture, software application is included for support, the cost and complexity increases. As the cloud is designed foremost to be economical for the user and provider, and servitisation is meant to fit a competitive business model, this clearly defines an issue for the servitization of this area.

Offering virtual environment testing as a service which covers a single use base, as a bespoke service, encounters fewer challenges, as the system complexity and cost is kept to a minimum. Conversely, when offering it is a cloud based service, it is necessary to cover an extremely large number of use cases which in turn involves a drastic increase in cost and complexity, upon the cloud providers' part.

Therefore, the ability to offer virtual infrastructure testing as a cloud service may be accomplished through the mitigation of this economic problem. As such the next section provides a breakdown and discussion of the challenges in order to derive future areas of research.

Overall, an analysis of both academic and industrial offerings in the area of cloud-based virtual environment testing as a service has shown that there is no application, environment or service offering which provides complete coverage of the required non-functional and functional testing types and that the area of evaluating cloud environments and using the cloud as a testing environment is under-researched. Therefore we can say that from an analysis of these sections there is a gap within both academia and industry in offering a full cloud-based service for testing virtualised environments. The next section will examine a variety of challenges posed to such a solution, which were derived form an analysis of this work.

V. CLOUD-BASED TESTING SERVITISATION

The challenges discovered within the previous section were found to be wide ranging and many. Many of these challenges were a result of the cloud environment, or the complex nature of the environments to be tested and hence the testing process as well. It was therefore realised that the lack of a complete solution for a cloud based virtual infrastructure testing service was due to the clash between the cloud-based servitisation of the complex testing environment. Therefore in order to mitigate this problem and successfully drive forward the solutions in this area, we propose a framework for cloud-based virtual infrastructure testing servitisation. Firstly, the requirements for a service are categorised through an analysis of the challenges noted in the previous section. Next, a framework for the service processes is presented, and then finally the requirements are mapped to the aforementioned process in order to complete the guidelines for producing a service.

Tremendous focus in software development has been trust towards evaluating requirements gathering techniques. An attempt to mention all works in this regard will result in an excessively long list, nonetheless, we refer to the reader to a survey [48]. An overview of such technique include model-driven techniques discussed in [49], group eliciting methods which include RAD/JAD workshops and prototyping techniques [50] which are generally used where there exists a significant amount of uncertainty about the requirements. In addition, knowledge-based management systems gave rise to cognitive [51] and contextual techniques [52], to mention a few.

Despite a huge research interest, we deduce that requirements gathering techniques are generally viewed as traditional or conventional, with a variance in opinions on methodological fundamental between advocates of each view. It is our view nonetheless that, the positives in each group of approaches are complimentary, and as such, we focus here on a method which suits the nature of this article. We use a scenario-based approach [53] which foremost, encapsulates significant enterprise-centric data we gathered in this study, and the secondly highlights the significance of the enterprise-driven method we will follows in the remainder of this article.

In this section, the servitization of software testing for virtual environments is illustrious of the enterprise organisation, with business goals, tasks, business rules of operation, and a clearly defined aim of the system. Premised on a high-level business goals which will be described in the case scenario to follow, we aim to demonstrate the high-level business goals which we deductively derive from an analysis in section **. In line with an argument in [54], it is our opinion that from high-level business goals, fine-tuned operational requirements can then be extracted by iteratively repeating the requirement gathering process, until operational requirements are met.

The testing target is considered to be a virtualisation-based corporate IT infrastructure of arbitrary size. The environments may consist or one, or a combination of some of the following examples:

- **VDI Infrastructure** - Virtual Desktop Infrastructures are becoming common place due to their economy and ease of management. Typically desktop machines will be replaced by thin-clients, which will connect to a hypervisor where the desktop (or applications) are stored and executed. These systems will likely be many, with small resource requirements.
- **Information Systems** - The systems which support corporate environments may also virtualised. These could include typical services such as storage, e-mail and web services. Or it might include servers for administration such as finances. These systems are likely to be fewer than the desktop infrastructure but with greater resources per unit, with scalable resources.
- **Resource Intensive** - These virtual instances are concerned with heavy resource intensive tasks, leveraging virtualisation architecture to perform heavy processing operations such as data mining or simulation. These machines vary in resource type but will often have one or two resources which are particularly and high and a large number of them.

A. VIRTUAL ENVIRONMENT TESTING AS A SERVICE: CHALLENGES

Cloud-based testing of virtual environments presents a number of important challenges, which could be grouped into three distinct categories; technical, business, legal and privacy. For brevity's sake, and to aid simplicity in our illustrations we assign each category a unique identification; T for technical, B for Business and L for legal.

Herein, the technical challenge group describes issues that broadly relate to the underlying software and hardware technologies used. These generally concern ineffective implementations which have a negative impact on the overall testing process. Technical issues may increase in intensity as they propagate through the employed methodology, creating poor environments for empirical testing by misrepresenting real world scenarios. Technical issues are easily identified through performance evaluation metrics. Solutions to these issue will typically require one or more of the following: Novel software or application implementations, and additional hardware or more efficient architecture design. Furthermore, the business challenge group describes issues regarding knowledge and processes required for the successful and economical design, execution, analysis and evaluation of the service offering. A variety of business stake-holders are involved in the process, and may be associated with a number of providers: Cloud Service Providers, Software Testing Service Providers, and Corporate Users of Virtual Infrastructure. Solutions for business issues involve optimizing methodologies and business processes including additional knowledge into the service, or periodically reviewing the efficacy of a process.

In addition, Legal and Privacy challenge group describes those issues which are typically concerned with any issues which involve a potential for deployment or use of the system to be in breach of legislation in any country. These issues are particularly inherited from the cloud computing paradigm, which is often criticized for concerns relating to legislation due to the variable state of data-centre location.

Software testing focused considerations for each challenge groups are described in detail in the following subsection.

1) TECHNICAL

a: T1 - SECURITY OF TEST DATA AND TEST ENVIRONMENT

Software testing is an activity that is often not considered as a business-critical activity. This makes it an ideal activity that can be moved to the cloud without fear of risks to clients' business critical data, for instance, undermine the integrity of. However, the test data and the test environment need to be adequately secured. [2], [3], [9].

b: T2 - NEED FOR ADEQUATE METRICS FOR MEASUREMENT OF RESOURCE USAGE

Considerations for ensuring availability of adequate metrics, tailored for analyzing usage of provisioned resources within cloud-based testing environments, while running tasks and processes. Metrics mechanism should be capable of issuing indicators/notifications such as: completion of test case execution, under-utilization or over-utilization of resources to trigger auto-decommissioning or auto-provisioning of more resources. In essence, this could be a context-aware metrics measurement mechanism that can be a triggered to an automated cloud manager, or be integrated within the functions of a cloud manager within a cloud-based testing platform [9], [55].

c: T3 - APPROPRIATE AND ACCURATE TEST DATA GENERATION

In some testing scenarios, effectiveness of testing may require actual production data. In cloud computing, privacy issues surrounding the ownership of data, and transparency in data handling is an existing issue [56]–[58]. But that still creates an area of concern when clients know that their data whether, mission-critical or not will still need to be 'out there' in the cloud for the sake of increasing testing effectiveness in the cloud. Therefore, cloud-based testing scenarios may need to come up with a solution that can accept production data, perform transformations on the content but retain similar structure to pass as mock production data that can be as close as possible to the real deal.

d: T4 - ARTIFICIAL FAULT GENERATION

In any test environment, the ability to generate artificial faults in a necessity [32], particularly in order to determine resiliency within theapplication. Fault injection is relatively simple for software based problems, but hardware based fault injection maybe more difficult. Although similar to test data generation, artificial fault generation exists at a different level and will need to be developed and managed appropriately.

e: T5 - CUSTOMIZATION OF TOOLS

In order to provide sufficient testing functionality for a variety of use cases, open-source tools are widely leveraged due to their minimal operating costs, high customizability and (in many cases) reputation for reliability, accuracy and stability [26], [34]. However, many of these tools are not standardized to suit any particular testing environment or framework. Therefore they may need to be customized, which will require development in expertise in numerous programming language. Development of testing will also require a generalized framework for integration of these tools into the overall testing framework.

f: T6 - MIGRATING NON-CLOUD APPS TO THE CLOUD

Some applications which are required to be tested within virtual environments are not developed to be executed upon the Cloud [2], [26]. Therefore one cited issue is that many applications may need to be adjusted for suitability within the cloud, taking advantage of and working in tandem with cloud features (scalability etc.). However this issue is not seen as too pertinent to the project due to the nature of environmental encapsulation offered by virtualized environments which are a key target for testing.

g: T7 - HEAVY RELIANCE UPON AUTOMATION

Automation is vital through the testing process for a number of reasons [2], [31], [32]. It allows tests to be streamlined by minimizing human input, maintains consistency between tests by ensuring stages are executed at the correct times and is overall necessary to operate highly complex environments such as the testing framework, and the targets of the test case. This will essentially require a considerable amount of development, with this development comes increased complexity. Therefore it is necessary to ensure that this automation is developed with stability in mind, so as to minimize errors within the system; as minimal as possible, so as to maintain low levels of complexity, and with efficiency in mind, so as to not delay the operation of the testing environment.

h: T8 - INTEROPERABILITY

Allowing cloud applications to operate upon different cloud software and virtualization platforms is an inherent issue to cloud computing [2]. Preventing 'vendor lock-in' is an issue solved by open container formats, open-source APIs and other mitigation efforts by the online community and software vendors. As such, is not seen as a huge issue within this project.

i: T9 - ACCURATE ENVIRONMENT REPRESENTATION

Nowadays, computer environments are not only complex, but are also formed from a huge variety of different software and hardware architectures. Ensuring these are included within the system is essential to operating an appropriate service. Additionally, the virtual environments used for testing may also not accurately represent those within production environments [30], if this is the case then testing process could be considered inaccurate. A key problem driving this issue is VM collocation interference, which as a known factor within virtualized environments.

j: T10 - ACCURATE MEASUREMENT

Providing accurate measurement is difficult under any circumstances as merely implementing measurement practices will cause additional strain upon the system [2], [3], [31]. It is therefore important to ensure that any measurement procedures are implemented correctly, consistency and minimally. Software automation and correct application of pre-existing APIs will enable this.

k: T11 - INHERENT CLOUD FEATURES

There are some features of cloud environments which may cause issues during testing [2], [3], [32]. For example, the ability of the cloud to scale enables multiple tests (with varying configurations) to run in parallel. However, scalability is costly, and therefore might cause unexpected financial burden upon the end user. This strengthens the argument for needing adequate and transparent pricing models. A feature which is fortunately already enabled within the cloud to a certain degree but some adaption will be necessary. Migration could be a potential issue during test cases as it would cause the environment to not be static and therefore cause measurement metrics to be inaccurate and difficult to repeat. This strengthens the argument for a greater degree of control over, or development of an entirely new, cloud management system for the testing process.

l: T12 - SOFTWARE DEVELOPMENT PROCESS INTEGRATION

The software development lifecycle varies form use case to use case, a service offering which includes the ability to conduct software, as well as system, testing must integrate effectively with the development life cycle [26]. As this may involve inclusion in a number of different places, and with manual processes, it is therefore essential that each integration occurs smoothly.

m: T13 - CLOUD ENVIRONMENT DEPENDABILITY

Due to the layered nature of the cloud, the dependability of it's lower layers may be unknown to a higher level service [59]. Therefore failures occurring at a low level may have an impact upon the testing process occurring at a higher level in that a low level fault may be recorded as an error in the testing of the virtual environment. Therefore a method of ensuring knowledge of the lower levels of cloud dependency within the system is necessary for an accurate testing process.

2) BUSINESS

a: B14 - NEED FOR ELABORATE PRICING MODELS

Cloud services have been widely reported to present cost-effective alternatives to traditional means of accessing and utilizing computing resources and services, amongst other benefits [29], [60]. But, there still remains a need to move away from pricing approaches showing only a high level view of prices for cloud services [61]. There is a need for more transparent pricing models showing more descriptive and detailed pricing for related services and service components, e.g. network-bandwidth costs.

b: B15 - SECURITY AND PRIVACY FOR BUSINESS

despite being, in the majority, a technical issue, requiring technical solutions; security may also be grouped under a business challenge. As data breaches may be costly to a business and an end user and therefore appropriate mitigation procedures must be balanced with risk in order to ensure the cost of a potential data breach is minimized [2].

c: B16 - LEVEL OF DOMAIN KNOWLEDGE REQUIRED

for each individual testing project undertaken, an appropriate level of domain knowledge will be required in order to carry out the test as accurately as possible. Where less than suitable knowledge exists for the test, results may be inaccurate or not easy to repeat. Certain situations such as testing the security of an application will undoubtable require expert domain knowledge in that area [9].

d: B17 - BALANCING BUSINESS CRITERIA WITH TESTING GOALS

one cited issue is the need to align testing criteria with business related goals. Many of these may be summarized as ensuring high economic output as a result from accurate testing [3]. For example, they cite the need to minimize errors, accurately predict the target, and high accuracy during reproducibility of tests.

e: B18 - AVAILABILITY OF TEST DATA

In some cases, test data may not even be available, which is certainly the case when legislation concerning sensitive issues comes into play [9]. Therefore may be necessary to derive a process for ensuring artificial test data accurately represents the data it replaces.

f: B19 - TESTER AVAILABILITY

In addition to automated test data, functional components will often need to be conducted by hand [2] and therefore require manual testers to be available. This can create problems during the testing process such as halting tests until the number of testers are available. This issue may be mitigated in part through the rise of online services such as amazon mechanical turk.

3) LEGAL AND PRIVACY

a: L20 TEST DATA PRIVACY

Privacy of end-user data is an issue that is inherent to the cloud [2]. It is related to security within cloud systems and their ability to withstand data breaches both from external and internal adversaries. The security of cloud environments is a contentious issue but is already typically solved by relevant technologies such as encryption and reliance upon trust between the CSP and the end-user. Service-Level-Agreements and accountability projects aid in mitigating and resolving issues that arise from security breaches.

b: L21 - LEGISLATION

As a CSP may host the physical location of their hardware in numerous countries, it is important to ensure all operations undertaken upon their systems are legal within the country that they are located in [2]. This can create issues due to the self-service nature of the cloud, allowing customers to provision resources anonymously and autonomously, in turn disallowing sufficient oversight by the CSP to ensure the relevant legislation is being adhered to. These issues are typically solved by service agreements between the two parties.

c: L22 - MALICIOUS AND ILLICIT USE

the autonomous and anonymous provisioning of cloud services ensures they are open to abuse [56]. A cloud service provider must protect themselves against breaches of relevant legislation. A testing service could be leveraged to perform automated reverse engineering to discover security flaws, or the complexity of the testing process could ensure that DoS attacks are easily executed through overloading of the system with an inefficient system set up.

TABLE 6. ViTaaS requirements mapping.

Requirement	Challenges
R1 - Automated Provision	T7, T12, B14, B16, B18, B19
R2 - Multi-Platform Support	T5, T6, T8, T12
R3 - Accurate / Complex Pricing	T10, T12, B14, B17, L21
R4 - Security	T1, T2, T3, T4, T9, T10, B14, B17, L21
R5 - Accurate Test Process	T1, T2, T3, T4, T9, T10, T11, B16, B17, B18, B19, L20
R6 - Adhering to legalislation	T1, B15, L20, L21, L23

d: L23 - SOFTWARE LICENSES

The requirement of adequate software licenses is a constraining factor for the testing process [34]. If test cases are to operate in parallel, then a suitable number of licenses will be necessary else legislation will be breached by the service provider.

B. REQUIREMENTS FOR CLOUD-BASED VIRTUAL INFRASTRUCTURE TESTING SERVICE

The analysis of literature within the area, produced a number of challenges to the production of a virtual infrastructure testing as a cloud based service. In order to help drive the development in this area through a more thorough understanding of the necessities of the system from a technical, business and legal level, requirement categories were dervied via an analysis of the aforementioned challenges.

Foremost, the primary requirements of the system was produced, which provide the ability to offer virtual infrastructure testing. Next, each challenge was examined to further understand the requirements for the system. For example, many challenges were related to the accuracy of the testing process (R5), so challenges whose mitigation would improve the testing process were grouped under this category. For example, T1 mentions to improve the security of the test environment, poor security might entail poor integrity of the environment and therefore ensuring integrity will provide a better guarantee of the testing process. Another example is in Providing support for multi-platforms and architectures (R2), which creates greater complexity and cost of the system, but also allows the service to be offered to a wider use base. T6 is grouped under this category as through the adaption of non-cloud apps to cloud environments, the system would provide wider ranging support for systems and therefore enhance this requirement. Due to the number of challenges, this process continued until all the features for each challenge were mapped into requirements of the system.

These are summarised below:

1) **Automated Provision** - The ability of a user to autonomously provision and execute tests.

2) **Multi-platform Support** - The system will support a wide variety of heterogenous virtualised software and hardware platforms.
3) **Accurate / Complex Pricing** - The requirement to autonomously generate accurate prices for the complex test products.
4) **Security** - The system should provide security for the test envrionment, test data, the system itsef and it's users.
5) **Accurate Test Process** - A test process should be conducted that is accurate and representative, adhering to empirical principles.
6) **Adhering to legislation** - The system should not breach any legilsation which covers the area the system is executed in.

The mapping of the specifc challenges to their requirements is given in table 7 and fig. 4.

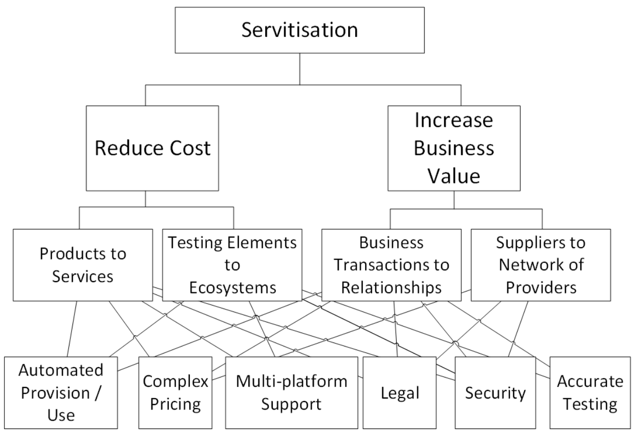

FIGURE 4. Service processes to requirements map.

C. SERVITISATION FRAMEWORK

Following on from the requirements derived in the previous section, the development of the cloud-based testing system is further extended through the proposal of a framework for servitisation. This framework was developed in order to provide an understanding on the required processes for migrating the service to the cloud, and therefore create dependencies between servitisation requirements from the previous section, and servitisation process. This is accomplished by first defining the business goals of the service, and then linking these goals to processes gathered from literature. What should be noted here is that, this framework is aimed at providing useful insights into knowledge to guide decision making. The framework is only limited to the context of our research, and thus requires a design solution to completely fit holistic testing. Nonetheless, the servitisation framework helps businesses by providing the processes for 1) companies to objectively assess their goals, 2) identifying key processes to follow in order to meet set goals, and 3) mapping servitisation requirements processes, and identifying dependabilities.

Traditional software testing, an inherent component of traditional software development is only appropriate for testing individual software components [62]. Software testing involves some generic and well-defined functional and non-functional processes that are commonly reflected, and are typically applicable across varying testing scenarios (refer to previous sections for more detail on testing). Although the technical processes of testing (functional and non-functional) drive the business transactions between organisations, for instance, a testing provider and a client, technical testing processes currently exist outside the testing provider-client boundary. The alternative is the concept behind service-oriented provisions, where flexible, dynamic business processes are created with enough agility to span across multiple organisations and multiple platforms [63]. Such provision can arguably be viewed synonymous with other utilities such as water, gas, electricity, etc. [64] That being the case, we postulate that cloud-based testing of VE can indeed evolve to be provided as basic level of service, at least. To that end, we propose a business-centric testing framework drivel by common testing requirements which enables and optimizes business functions such as pricing, business relations, service agreements, legal liabilities, and networking with providers. In this manner, business functions of testing can be blended with other testing requirements such as quality assurance, security and integrity of data [65]. The vision is to provide a strategy that is also customer-centric, providing competent market-based resource management in both computing risks and service agreement [66]. Ultimately, it is also about adding value to a business while elevating its competitive edge, and creating new relationships [67].

We thus define cloud-based software testing servitization as: *the process of creating complex ecosystems of testing services from networked providers, providing software testing services instead of testing products, and building business relationships instead of business transactions.* We argue that, with servitization, it is indeed plausible to reduce costs while increasing the business value of testing.

1) GOALS

In our proposed framework in fig. 5, goals for servitization include processes and strategies necessary to moving up the value chain, focusing on providing sophisticated services [68]. Increasing business value and reducing costs, distinguish the first layer (what we propose as goals) of our proposed framework in line with Vandermerwe and Rada's servitization concept [67].

2) PROCESSES

Functional processes include transforming products to services, elements to ecosystems, transactions to relationships, and suppliers to networked providers, describes the transformations of ''how what goals are to be achieved'' to ''how to achieve the goals''.

D. SERVITISATION PROCESSES MAPPING

Finally, to complete the service definition, the requirements for the cloud based framework are mapped to the service processes. This then allows a clearly defined understanding of the way in which improving the aforementioned requirements will support the processes necessary for servitisation. Mitigation of the listed challenges for each requirement will therefore aid in the development of this process.

1) TRANSFORMING PRODUCTS TO SERVICES

In order to create enhanced value and capacity, it is essential that a process enables changes, to ensure that the cloud-based software testing product is able to be delivered a service. Thus, the process of transforming *products to services* provides the following:

- Automated Use in order to allow service access
- Complex Pricing in order to manage charging for the service
- Multi-platform support to allow a wide user base
- Sound legal basis to ensure the service is legitimate
- Security to provide and ensure trust in the system

2) TRANSFORMING TESTING ELEMENTS TO ECOSYSTEMS

The second step towards reducing costs encompasses processes for converting traditionally individual *testing elements into ecosystems of services*. Along with ensuring that test data and the test environment are secure, the following requirements are important:

- Automated use to allow complex components to operate
- Multi-platform support to facilitate a wide range of testing services
- Security to ensure the integrity of the testing process
- Accurate testing to ensure that the testing process is effective

3) TRANSFORMING BUSINESS TRANSACTIONS TO RELATIONSHIPS

A servitization strategy entails that in addition to consolidating the core services, it it paramount to have a process in place, which focuses on creating intimate relationships with clients and suppliers. Thus, in order to transform *business transactions into relationships* the following requirements ought to be met:

- Automated use to facilitate that the service is operated on an on-going basis
- Complex pricing to accommodate and facilitate a variety of use cases
- Legal requirements which ensure that all provided services are legitimate
- Security, to provide trust between partners
- Accurate testing which will ensure that relationships continue and strengthen

4) TRANSFORMING SUPPLIERS TO NETWORKS OF PROVIDERS

Business opportunities such as marketing whereupon value creation is a result of cooperation between suppliers and customers, encompasses considerations for a changes in risk ownership, information sharing and a consistent delivery of information among partnerships. Thus, in order to transform *suppliers into networks of providers* the following service requirements ought to be considered:

- Complex pricing to facilitate a variety of goals
- Legal processes to provide a legitimate frameworks for negotiations.
- Security to provide trust and strengthen relationships

The mappings of these relationships is presented in fig. 6. Finally, in order to help drive the development of this framework, the next section proposes an analysis and some research trends to aid in enhancing the aforementioned requirements.

E. ANALYSIS

In this section, we briefly describe an individual organisation's servitization requirements in order to provide context for out analysis, and direction for future areas of research.

1) CASE SCENARIO

In order to provide a context and basis for the process of servitisation, the following real-world case study will be evaluated against the challenges posed in the previous section, in order to derive the necessary requirements for the process.

The case-study in question is concerned with the development and implementation of a public-cloud service which permits a customer to evaluate and optimise their virtual environments. Within this context, virtual environment refers to the collection of virtual machines, virtual appliances and their underlying supporting architecture i.e. hypervisors or container-based virtualisation technologies. The following are requirements for Company X

- Static Usage Cost of £č100 to £č200 hourly
- Subscription Options dependant of service
- Network Performance to verify sufficient capacity
- Load Testing to assess response time under varying usage levels
- Stress Testing to determine minimum and maximum boundaries for system usage
- Online Help Centre for facilitate discussions between users and the support team
- A ticketed system which provides efficient support from an external team
- Initial and on-going training service as software receives updates or changes
- Multiple Client Support including a variety of device types

The ability to provide automation in the servitized system is a necessity due to the nature of cloud services, and the complexity of the testing process. However implementing automation can be challenging in complex processes, whilst the automation of the cloud can have a negative impact on emprical processes due to the increased dynamism. A primary area of focus we propose to mitigate this is through the development of a custom cloud management software, or heavy adaptions of the cloud manager in use.

Pricing models required of the system have been considered to complex due to the occasionally non-deterministic nature of the testing process. A number of areas of focus for research would benefit this issue. Firstly, by understanding and employing more accurate metrics, the costing methods can become more realistic. The application of predictive analytics/forecasting to the testing process, will ensure that pricing quotes are more accurate and finally, through providing greater system performance optimisation, such as through optimisation of VM placement, the system can overally function more economically.

In regards to increasing wider ranging support, the main solution is a costly one,i.e. to simply integrate greater diversity and redundancy into the system. Therefore a cost-benefit analysis employed by the service provider to examine which particular users might use the service more frequently would be beneficial.

Indeed, servitization comes with its legal considerations for enforcing and assuring quality, while reinforcing responsibility and accountability in business processes. Hence, a legal framework could be a central theme in future work to ensure that, for instance, there are common fronted assurances for all automated services regardless of a jurisdiction, but maintaining heterogeneity in whatever back-end automation services to fulfil different legal requirements. Additionally, with data centres built on land and in different jurisdictions, there is need for the legal framework to ensure that the servitization model is both enabled and yet complies with respective legislative discrepancies. This includes legal basis for defining imperatives about ownership and custodianship of, for instance, test data among the network of providers. Finally, a legal basis for control and utilization of the testing ecosystem will ensure fair business practices, particularly with pricing models.

The security of the cloud is an already a contentious issue, however the application of certain features such as cryptography, as well as research into enhanced techniques such as homomorphic encryption can improve customer trust in this area.

The area of improving adherence to empirical process is difficult to ensure under any circumstances. However, through providing accurate metrics, developing the custom cloud, increasing the accuracy of fault generation and integrating understanding of the cloud environment dependability within lower layers, accuracy can be improved. This area contains many potential solutions, most of which the service should select in accordance with an associated use-case.

VI. CONCLUSION

Virtual environments are beginning to underpin computing systems on a global scale, therefore testing these environments is essential to ensuring the economisation, and stability of computer systems. Unfortunately the complexity of modern networked environments has caused the evaluation of these systems to be highly expensive. To address this problem, this paper reviewed work in the area of VE testing, particularly the efficacy of leveraging cloud-based systems for automated testing as a service. A number of challenges were discovered from this review, which gave way to the development of some requirements for a servitisation of virtual environment testing. Furthermore, this extended into our developing a framework for servitisation. A number of further research areas were then highlighted, which would help pave the way for further research and possible systems development.

REFERENCES

[1] J. Sahoo, S. Mohapatra, and R. Lath, "Virtualization: A survey on concepts, taxonomy and associated security issues," in *Proc. 2nd Int. Conf. Comput. Netw. Technol.*, Apr. 2010, pp. 222–226.

[2] L. Riungu, O. Taipale, and K. Smolander, "Research issues for software testing in the cloud," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Technol. Sci.*, Nov. 2010, pp. 557–564.

[3] P. Robinson and C. Ragusa, "Taxonomy and requirements rationalization for infrastructure in cloud-based software testing," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Technol. Sci.*, Nov. 2011, pp. 454–461.

[4] H. Do, S. Elbaum, and G. Rothermel, "Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact," *Empirical Softw. Eng.*, vol. 10, no. 4, pp. 405–435, 2005.

[5] M. Utting, A. Pretschner, and B. Legeard, "A taxonomy of model-based testing approaches," *Softw. Test., Verification Rel.*, vol. 22, no. 5, pp. 297–312, 2012.

[6] M. Voznak and J. Rozhon, "SIP infrastructure performance testing," in *Proc. 9th WSEAS Int. Conf. Telecommun. Inform.*, May 2010, pp. 153–158.

[7] R. S. Pressman, *Software Engineering: A Practitioner's Approach*. Basingstoke, U.K.: Palgrave Macmillan, 2005.

[8] G. Denaro, A. Polini, and W. Emmerich, "Early performance testing of distributed software applications," in *Proc. 4th Int. Workshop Softw. Perform.*, Jan. 2004, vol. 29, no. 1, pp. 94–103.

[9] L. M. Riungu, O. Taipale, and K. Smolander, "Software testing as an online service: Observations from practice," in *Proc. 3rd Int. Conf. Softw. Test., Verification, Validation Workshops*, Apr. 2010, pp. 418–423.

[10] A. Bertolino, "Software testing research: Achievements, challenges, dreams," in *Proc. Future Softw. Eng.*, May 2007, pp. 85–103.

[11] B. Cao, Z. Chen, H. Liu, G. Ma, P. Zhang, and G. Peng, "Black box testing for cloud-based client security software in network behaviors," in *Proc. 4th Int. Conf. Netw. Distrib. Comput.*, Dec. 2013, pp. 75–79.

[12] F. Weidong and X. Yong, "Cloud testing: The next generation test technology," in *Proc. 10th Int. Conf. Electron. Meas. Instrum.*, vol. 2, Aug. 2011, pp. 291–295.

[13] E. Engström, P. Runeson, and M. Skoglund, "A systematic review on regression test selection techniques," *Inf. Softw. Technol.*, vol. 52, no. 1, pp. 14–30, Jan. 2010.

[14] J. Cleland-Huang and D. Schmelzer, "Dynamically tracing non-functional requirements through design pattern invariants," in *Proc. Workshop Traceability Emerg. Softw. Eng., Conjunction IEEE Int. Conf. Automated Softw. Eng.*, Oct. 2003, pp. 1–10.

[15] N. Mehrotra, "Cloud testing vs. testing a cloud," in *Proc. 10th Annu. Int. Softw. Test. Conf.* Bengaluru, India: Infosys, Nov. 2010, pp. 22–23.

[16] K. Ye, X. Jiang, S. Chen, D. Huang, and B. Wang, "Analyzing and modeling the performance in Xen-based virtual cluster environment," in *Proc. IEEE 12th Int. Conf. High Perform. Comput. Commun. (HPCC)*, Sep. 2010, pp. 273–280.

[17] R. S. Renner, B. M. Velichkovsky, and J. R. Helmert, "The perception of egocentric distances in virtual environments—A review," *ACM Comput. Surv.*, vol. 46, no. 2, p. 23, Nov. 2013.

[18] I. Foster and C. Kesselman, *The Grid 2: Blueprint for a New Computing Infrastructure*. Amsterdam, The Netherlands: Elsevier, 2003.

[19] R. Parmelee, T. I. Peterson, C. C. Tillman, and D. J. Hatfield, "Virtual storage and virtual machine concepts," *IBM Syst. J.*, vol. 11, no. 2, pp. 99–130, Jun. 1972.

[20] B. Jacob and T. Mudge, "Virtual memory: Issues of implementation," *Computer*, vol. 31, no. 6, pp. 33–43, Jun. 1998.

[21] A. T. Campbell, J. Vicente, and D. A. Villela, "Virtuosity: Performing virtual network resource management," in *Proc. 7th Int. Workshop Qual. Service*, Jun. 1999, pp. 65–76.

[22] K. Ye, Z. Wu, B. B. Zhou, X. Jiang, C. Wang, and A. Zomaya, "Virt-B: Towards performance benchmarking of virtual machine systems," *IEEE Internet Comput.*, vol. 18, no. 3, pp. 64–72, May 2014.

[23] K. Ye, J. Che, Q. He, D. Huang, and X. Jiang, "Performance combinative evaluation from single virtual machine to multiple virtual machine systems," *Int. J. Numer. Anal. Model.*, vol. 9, p. 2, May 2012.

[24] S. Nambisan and R. A. Baron, "Virtual customer environments: Testing a model of voluntary participation in value co-creation activities," *J. Product Innov. Manage.*, vol. 26, no. 4, pp. 388–406, Jul. 2009.

[25] C. Cadar, P. Godefroid, S. Khurshid, C. S. Păsăreanu, K. Sen, N. Tillmann, and W. Visser, "Symbolic execution for software testing in practice: Preliminary assessment," in *Proc. 33rd Int. Conf. Softw. Eng.*, May 2011, pp. 1066–1071.

[26] C. H. Kao, S. T. Liu, and C. C. Lin, "Toward a cloud based framework for facilitating software development and testing tasks," in *Proc. IEEE/ACM 7th Int. Conf. Utility Cloud Comput.*, Dec. 2014, pp. 491–492.

[27] T. Hanawa, T. Banzai, H. Koizumi, R. Kanbayashi, T. Imada, and M. Sato, "Large-scale software testing environment using cloud computing technology for dependable parallel and distributed systems," in *Proc. 3rd Int. Conf. Softw. Test., Verification, Validation Workshops*, Apr. 2010, pp. 428–433.

[28] L. Ciortea, C. Zamfir, S. Bucur, V. Chipounov, and G. Candea, "Cloud9: A software testing service," *SIGOPS Oper. Syst. Rev.*, vol. 43, no. 4, pp. 5–10, Jan. 2010. doi: 10.1145/1713254.1713257.

[29] T. King and A. Ganti, "Migrating autonomic self-testing to the cloud," in *Proc. 3rd Int. Conf. Softw. Test., Verification, Validation Workshops*, Apr. 2010, pp. 438–443.

[30] D. Citron and A. Zlotnick, "Testing large-scale cloud management," *IBM J. Res. Dev.*, vol. 55, no. 6, pp. 387–396, Dec. 2011. doi: 10.1147/JRD.2011.2170913.

[31] J. Gao, P. Pattabhiraman, X. Bai, and W. Tsai, "SaaS performance and scalability evaluation in clouds," in *Proc. IEEE 6th Int. Symp. Service Oriented Syst. (SOSE)*, Dec. 2011, pp. 61–71.

[32] Y. Shi, D. Zou, W. Qiang, X. Liao, and H. Jin, "CloudTB: A quick and reliable testbed for virtual machine based cloud computing systems," in *Proc. Int. Conf. Electron., Commun. Comput.*, Feb. 2015, pp. 40–47.

[33] M. Lynch, T. Cerqueus, and C. Thorpe, "Testing a cloud application: IBM SmartCloud inotes: Methodologies and tools," in *Proc. 2013 Int. Workshop Test. Cloud*, Jul. 2013, pp. 13–17. doi: 10.1145/2489295.2489299.

[34] H. Tómasson and H. Neukirchen, "Distributed testing of cloud computing applications using the TTCN-3-based Jata test framework," in *Proc. 2nd Nordic Symp. Cloud Comput. Internet Technol.*, Sep. 2013, pp. 22–29. doi: 10.1145/2513534.2513540.

[35] S.-H. Lim, J. Horey, Y. Yao, E. Begoli, and Q. Cao, "Performance implications from sizing a VM on multi-core systems: A data analytic application's view," in *Proc. IEEE Int. Symp. Parallel Distrib. Process., Workshops Phd Forum*, May 2013, pp. 1001–1008.

[36] M. Mas'ud, A. H. Yaacob, and N. Ahmad, "Network performance testing on VM based autonomous Web server," in *Proc. Int. Conf. Comput. Inform.*, Jun. 2006, pp. 1–6.

[37] H. A. Jabry, L. Liu, Y. Zhu, and J. Panneerselvam, "A critical evaluation of the performance of virtualization technologies," in *Proc. 9th Int. Conf. Commun. Netw. China*, Aug. 2014, pp. 606–611.

[38] G. Martinović, J. Balen, and S. Rimac-Drlje, "Impact of the host operating systems on virtual machine performance," in *Proc. 33rd Int. Conv. (MIPRO)*, May 2010, pp. 613–618.

[39] R. Chi, Z. Qian, and S. Lu, "Be a good neighbour: Characterizing performance interference of virtual machines under xen virtualization environments," in *Proc. 20th IEEE Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2014, pp. 257–264.

[40] Q. Zhu, J. Zhu, and G. Agrawal, "Power-aware consolidation of scientific workflows in virtualized environments," in *Proc. ACM/IEEE Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2010, pp. 1–12.

[41] Y. Koh, R. Knauerhase, P. Brett, M. Bowman, Z. Wen, and C. Pu, "An analysis of performance interference effects in virtual environments," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw.*, Apr. 2007, pp. 200–209.

[42] Y. Hashimoto and K. Aida, "Evaluation of performance degradation in HPC applications with VM consolidation," in *Proc. 3rd Int. Conf. Netw. Comput.*, Dec. 2012, pp. 273–277.

[43] X. Pu, L. Liu, Y. Mei, S. Sivathanu, Y. Koh, and C. Pu, "Understanding performance interference of I/O workload in virtualized cloud environments," in *Proc. IEEE 3rd Int. Conf. Cloud Comput.*, Jul. 2010, pp. 51–58.

[44] O. Tickoo, R. Iyer, R. Illikkal, and D. Newell, "Modeling virtual machine performance: Challenges and approaches," *SIGMETRICS Perform. Eval. Rev.*, vol. 37, no. 3, pp. 55–60, Dec. 2010. doi: 10.1145/1710115.1710126.

[45] S. Kundu, R. Rangaswami, A. Gulati, M. Zhao, and K. Dutta, "Modeling virtualized applications using machine learning techniques," *SIGPLAN Not.*, vol. 47, no. 7, pp. 3–14, Mar. 2012. doi: 10.1145/2365864.2151028.

[46] V. Soundararajan, B. Agrawal, B. Herndon, P. Sethuraman, and R. Taheri, "Benchmarking a virtualization platform," in *Proc. IEEE Int. Symp. Workload Characterizat. (IISWC)*, Oct. 2014, pp. 99–109.

[47] S. Jones, "Servitization: Lessons learned from the manufacturing industry," Black Pepper Software, U.K., Tech. Rep., Dec. 2012. [Online]. Available: https://www.blackpepper.co.uk/blog/servitization-lessons-learned-from-the-manufacturing-industry

[48] D. Zowghi and C. Coulin, "Requirements elicitation: A survey of techniques, approaches, and tools," in *Engineering and Managing Software Requirements*. New York, NY, USA: Springer, 2005, pp. 19–46.

[49] A. van Lamsweerde, R. Darimont, and E. Letier, "Managing conflicts in goal-driven requirements engineering," *IEEE Trans. Softw. Eng.*, vol. 24, no. 11, pp. 908–926, Nov. 1998.

[50] A. M. Davis, "Operational prototyping: A new development approach," *IEEE Softw.*, vol. 9, no. 5, pp. 70–78, Sep. 1992.

[51] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge engineering: Principles and methods," *Data Knowl. Eng.*, vol. 25, nos. 1–2, pp. 161–197, 1998.

[52] S. Viller and I. Sommerville, "Social analysis in the requirements engineering process: From ethnography to method," in *Proc. IEEE Int. Symp. Requirements Eng.*, Jun. 1999, pp. 6–13.

[53] A. Sutcliffe, "Scenario-based requirements engineering," in *Proc. 11th IEEE Int. Requirements Eng. Conf.*, Sep. 2003, pp. 320–329.

[54] B. Nuseibeh and S. Easterbrook, "Requirements engineering: A roadmap," in *Proc. Conf. Future Softw. Eng.*, Jun. 2000, pp. 35–46.

[55] L. Yu, W.-T. Tsai, X. Chen, L. Liu, Y. Zhao, L. Tang, and W. Zhao, "Testing as a service over cloud," in *Proc. 5th IEEE Int. Symp. Service Oriented Syst. Eng.*, Jun. 2010, pp. 181–188.

[56] T. Dillon, C. Wu, and E. Chang, "Cloud computing: Issues and challenges," in *Proc. IEEE Int. Conf. Adv. Inform. Netw. Appl.*, Apr. 2010, pp. 27–33.

[57] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Services Appl.*, vol. 1, no. 1, pp. 7–18, 2010. doi: 10.1007/s13174-010-0007-6.

[58] Y. Jadeja and K. Modi, "Cloud computing - concepts, architecture and challenges," in *Proc. Int. Conf. Comput., Electron. Elect. Technol. (ICCEET)*, Mar. 2012, pp. 877–880.

[59] Y. Pan and N. Hu, "Research on dependability of cloud computing systems," in *Proc. IEEE ICRMS*, Aug. 2014, pp. 435–439.

[60] A. Younge, R. Henschel, J. Brown, G. von Laszewski, J. Qiu, and G. Fox, "Analysis of virtualization technologies for high performance computing environments," in *Proc. IEEE 4th Int. Conf. Cloud Comput.*, Jul. 2011, pp. 9–16.

[61] J. Zhou, S. Li, Z. Zhang, and Z. Ye, "Position paper: Cloud-based performance testing: Issues and challenges," in *Proc. Int. Workshop Hot Topics Cloud Services* 2013, pp. 55–62. doi: 10.1145/2462307.2462321.

[62] M. N. Huhns and M. P. Singh, "Service-oriented computing: Key concepts and principles," *IEEE Internet Comput.*, vol. 9, no. 1, pp. 75–81, Jan. 2005.

[63] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-oriented computing: State of the art and research challenges," *Computer*, vol. 40, no. 11, pp. 38–45, Nov. 2007.

[64] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generat. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, 2009.

[65] M. P. Papazoglou and B. Kratz, "Web services technology in support of business transactions," *Service Oriented Comput. Appl.*, vol. 1, no. 1, pp. 51–63, Apr. 2007.

[66] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud Computing: Vision, hype, and reality for delivering IT services as computing utilities," in *Proc. 10th IEEE Int. Conf. High Perform. Comput. Commun.*, Sep. 2008, pp. 5–13.

[67] S. Vandermerwe and J. Rada, "Servitization of business: Adding value by adding services," *Eur. Manage. J.*, vol. 6, no. 4, pp. 314–324, 1989.

[68] Z. Ahamed, T. Inohara, and A. Kamoshida, "The servitization of manufacturing: An empirical case study of IBM corporation," *Int. J. Bus. Admin.*, vol. 4, no. 2, p. p18, 2013.