# Position Paper: Cloud-based Performance Testing: Issues and Challenges

Junzan Zhou, Shanping Li, Zhen Zhang, Zhen Ye
College of Computer Science and Technology
Zhejiang University
Hangzhou, China
{zhoujunzan,shan,zhen,yezhen}@zju.edu.cn

## ABSTRACT

Conducting performance testing is essential to evaluate system performance. With the emergence of cloud computing, applying cloud resources for large-scale performance testing become very attractive. Many organizations have applied cloud-based performance testing in realistic projects. Cloud computing brings many benefits for performance testing, while we also have to face many new problems such as performance variation of cloud platform and security problems. In this overview, we discuss the differences between traditional and cloud-based performance testing. We investigate the state-of-art of cloud-based performance testing. We address the key issues with relevant challenges. For some of the issues, we formalize the problems and give our initial idea. We focus on the quality of workload generation and present our experimental results to validate the existence and degree of the challenges. We think that it is beneficial to apply cloud-based performance testing in many cases.

## Categories and Subject Descriptors

D.2.5 [**Software**]: Testing and Debugging

## Keywords

cloud, performance testing, load testing, challenge, overview

## 1. INTRODUCTION

Performance, scalability and reliability are critical properties for many systems such as financial systems, end-user services, or computing and networking infrastructures. However, research showed that more than half of the organizations suffered from performance issues [12]. Performance testing is an essential way to reveal performance problems ranging from poor throughput to system inconsistency or crash. Other techniques are possible, based on performance modeling and simulation, to predict a system performance and capacity. These techniques can be complementary under some circumstances.


*HotTopiCS'13,* April 20–21, 2013, Prague, Czech Republic.


Performance testing is a resource intensive activity. Traditional LAN-based performance testing needs to host a large amount of machines for simulating thousands of users. With the emergence of cloud computing, it is possible to use cloud resources for large-scale performance testing. The way of pay-as-you-consume enables organizations to conduct performance testing without hosting costly infrastructures.

Many organizations have launched their products of cloud-based performance testing. Recently, performance engineers from SOASTA [4], a leader of cloud testing, ran extensive performance tests for NASA's Curiosity Rover to prepare for hundreds of GB of streaming video to be beam backed to Earth [32]. With the convenience provided by cloud computing, service providers can deliver their performance testing services more easily than before. Cloud-based performance testing services enable organizations to use online services without employing a group of performance testing engineers and host performance testing infrastructures. It is especially attractive to small and medium companies that lack performance testing professionals and budget.

Cloud-based performance testing is quite different with traditional LAN-based performance testing. The merits introduced by cloud computing are obvious, but we also need to face many new challenges in the meanwhile. First, we need to take into account the measurement overhead and performance variation of cloud platforms to insure the quality of testing results. Next, from the perspective of cloud computing consumers, it's imperative to figure out a cost model for conducting performance testing once or repeatedly. Then, other issues such as information security of testing scripts and SLA of performance testing are also very important for performance testing service providers. Thus, performance testing need to tackle many new challenges when migrating to clouds.

In this paper, we discuss the issues and challenges of migrating performance testing to cloud. Issues, such as quality of workload generation, security, cost and SLA, are discussed. We compare the experimental results of workload generation in LAN with those in EC2, which shows that the quality of workload generation in EC2 is potentially less stable. The contributions of this paper are not only a vision of cloud-based performance testing, but also a list of research challenges that need to be addressed in further research. For some of the challenges, we briefly propose our initial ideas of approaching them.

The rest of this paper is organized as follows. In section 2, we briefly visit relevant concepts of performance testing. We discuss the differences between LAN-based and cloud-based

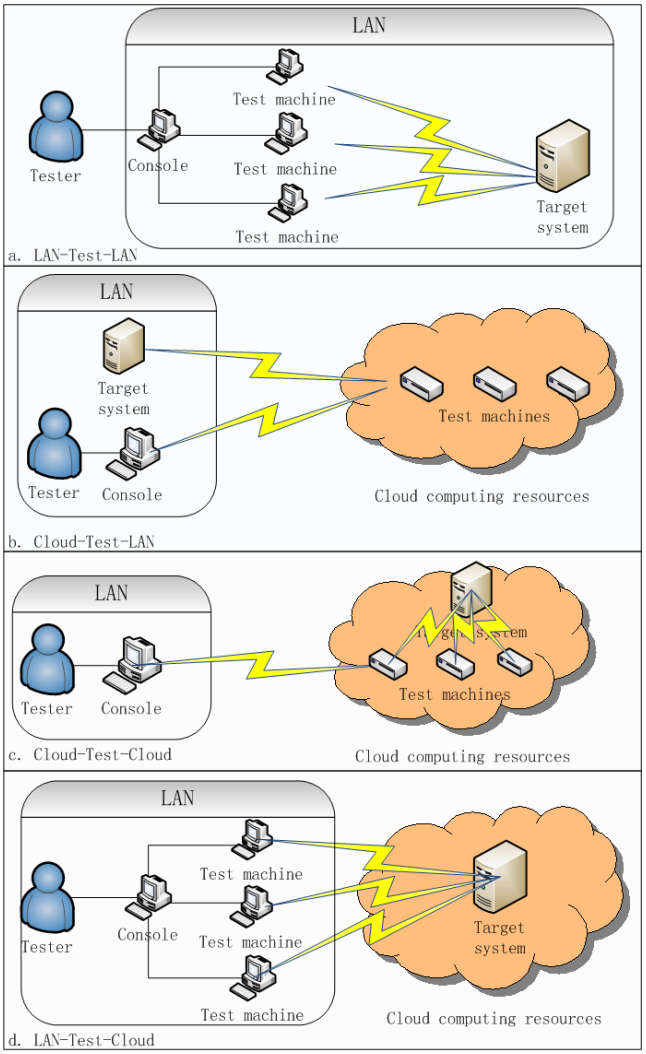


**Figure 1: Deployments of performance testing**

performance testing. In section 3, we present and discuss related works. Then, we demonstrate the issues and challenges in section 4. We formalize some of the problems and give our initial ideas. In section 5, we show our experimental evaluation of workload generation on EC2 by comparing with that in LAN. Finally, we give the conclusion remarks and our working-in-progress in section 6.

## 2. OVERVIEW OF CLOUD-BASED PERFORMANCE TESTING

Performance testing is a subset of both software testing and performance engineering. Generally, performance testing includes load testing, stress testing, availability testing, scalability testing and so on. The boundary between different kinds of performance testing, like stress testing and load testing, is flexible and dependent on the purpose of performance testing. Generally, load testing, stress testing and availability testing are regarded as black-box methods, while performance profiling and unit performance testing are known as white-box methods. Cloud-based performance testing aims at making the best use of the elastic cloud resources. In this paper, the term "performance testing" refers to black-box methods.

Cloud-based performance testing applies cloud resources for producing synthetic workloads. Performance testing is suitable to be migrated to cloud. Figure 1 shows four possible deployments of performance testing. The first topology is the traditional one (LAN-Test-LAN), which deploys both testing agents and target systems in LAN. The latter two deployments are cloud-based performance testing, which deploy test agents on cloud and target systems inside or outside cloud. Testers can control the test machines on cloud through a console to manipulate all test activities. The last one, LAN-Test-Cloud, should *not* be considered as cloud-based performance testing. Therefore, from the view of deployment, cloud-based performance testing can be: (1) Cloud-Test-LAN, (2) Cloud-Test-Cloud.

The differences between cloud-based and LAN-based performance testing are related to their deployment environment. Infrastructure, software architectures and business models are quite different between LAN and cloud, which lead to at least the following diversities and issues:

- Utilization of computing resources. Traditional performance tests generally deploy test agents on distributed physical machines or test clusters that are in the same sub-net. Cloud-based performance testing might deploy test agents on virtual machines provided by IaaS providers like EC2, which means: 1) each test agent has a certain possibility of sharing physical resources with other test agents; 2) each test agent would share physical resources with other applications, 3) the execution of tests and measure are influenced by scheduling of both OS and hypervisor. These potentially render the decrease of test quality and measurement accuracy.

- Location of test agents. Deploying test agents on a LAN makes the delay from test agents to target system to be shorter and more stable than on clouds. Therefore, cloud-based performance testing would be more likely to suffer from fluctuation of network delay. This issue should be taken into account when designing performance testing strategy, conducting workload characterization and analyzing test data.

- Cost. The expenditure of traditional performance testing can include but not limited to: 1) salary of engineers; 2) cost of purchasing and maintaining infrastructure; 3) cost of licensing and services. However, the expenditure of cloud-based performance testing can include but not limited to: 1) salary of engineers; 2) cost of cloud pay-as-you-consume resources; 3) cost of licensing and services. It is obvious that the main difference is the cost of testing infrastructure. If customers purchase testing services from performance testing service providers, the structure of cost mainly depends on pricing of services.

- Security concerns. The security concerns are related to the privacy of resources, since LANs are usually for private access, while public clouds are open to users. Therefore, more security problems are needed to consider for applications like performance testing. The security concerns of cloud-based performance testing are different for diverse stakeholders. For organizations

that apply cloud computing for performance testing, they may concern about the security of sensitive information contained in test scripts. For service providers of cloud-based performance testing, they may need to avoid hackers utilizing cloud resources for large-scale DoS attack. These issues are different when testing in LAN, which may have been ignored in most cases.

# 3. RELATED WORK

Most related work studied cloud-based software testing, only a few focused on cloud-based performance testing. Hu and Xiao discussed the advantages and disadvantages, opportunities and challenges of cloud-based automatic software testing, but did not focus on cloud-based performance testing [23]. Kim et al. examined the issues that hinder rapid adoption of cloud computing [15], which gave some reference when adopting performance testing in cloud. Candea, Bucur, and Zamfir discussed three categories of Test as a Service (TaaS) and challenges of TaaS [17]. And later, they published a TaaS platform called Cloud9, which is designed to run as a web service based on cloud [7]. Hanawa et al. proposed a cloud-based software testing environment named D-cloud, which is mainly used to reproduce hardware faults for dependable parallel and distributed systems [14]. T. King and A. Ganti showed their work of migrating autonomic self-testing to the cloud [18]. Riungu et al. conducted an objective qualitative study to identify the conditions that influence software as an online service and elicit some issues of software testing [28]. We think it is a good reference for those who are going to do testing in cloud. Zohar et al. described an methodology of conducting performance testing for a network management system based on Amazon AWS [13]. Snellman, Ashraf and Porres shared their experience of implementing performance testing for rich internet applications in cloud [31]. Jun Wang and Fanpeng Meng tried to answer the two questions of cloud testing: (1) Why do cloud-based testing, (2) How to do cloud-based testing. However, it's only a general guide, and they did not cover many problems like costing models and constraints of cloud [16].

Although there are just a few papers depicting performance testing experience based on cloud, the industrial practitioners like SOASTA [4] and HP [1] have already released their cloud-based performance testing products. Their products and related documents show that they are trying to harness the benefits of cloud computing and developing relevant tools. Many other companies, such as Load Storm, Keynote, and Cloud Assault, are also devoting much effort to migrating performance testing to public clouds.

# 4. ISSUES AND CHALLENGES

In this section, we detail the key issues and relevant challenges of cloud-based performance testing in Table I. We focus on the issue of quality of workload generation which we regard as the most critical issue.

## 4.1 Quality of workload generation

The quality of workload generation is very important for producing convincing performance testing results. Otherwise, test results will be unbelievable and cause waste of test resources. On a cloud platform, the quality of performance testing can be influenced by two key factors: 1) over-

**Table 1: Issues of cloud-based performance testing**

| Issues | Issues of cloud-based performance testing |
|---|---|
| | Description |
| Quality | How to control performance testing quality? |
| Data Analysis | How to analyze performance testing results? |
| Security | How to keep test data secure?<br>How to avoid services being misused? |
| Cost | How much will a test need?<br>How many resources are needed for a test? |
| SKA | How to ensure SLA of performance testing? |

much workload on test machines; 2) performance variation of cloud. Other factors such as scheduling of hypervisor, scheduling of OS, and overhead of instrumentation may also introduce measure errors. In order to generate qualified performance testing results, we have to give solutions to at least the following questions: 1)what kind of results are convincing? 2) how to to figure out the capacity of different kinds of instances for different workloads? 3) how to adaptively control the distribution of workload generation locally and globally?

### 4.1.1 Metrics for performance testing quality

For assessing the performance testing quality, new metrics are needed to denote the quality of the synthetic workload generation and measurements. The metrics should help us to distinguish whether the generation of synthetic workload is good or bad. Although the issue of workload generation quality has been pointed out more than fifteen year ago , to the best of our know, there is still no metrics for quantitatively to denote the quality of workload generation. In order to quantitatively evaluate the quality of workload generation, we defined two metrics called "think time deviation" and "degree of workload deviation" to assess the quality of workload generation. Think time deviation is used so assess the precision of execution from the last execution in terms of time interval. The metrics can help reveal the degree of the thread starvation and precision of thread scheduling. Think time deviation can only address the quality of workload generation for some specific threads, but not address the global workload generation quality. Therefore, degree of workload deviation is introduced to address the global workload generation quality as a complementary. We identify whether a generation is outlier based on think time deviation. Then, the degree of workload generation indicate the percentage of outliers in overall process of synthetic workload generation. For the limitation of the space, we will demonstrate the metrics in another paper.

### 4.1.2 Capacity of performance testing machines

Capacity planning can be used to help us understand the capacity of different test machines. Then, the amount of instances that are needed for a cloud-based performance test can be further figured out. In the conventional context of capacity planning, capacity is the maximum amount of tasks that a system can complete during a given interval. In this case, the goal of capacity planning is to predict the minimum number of machines for completing a performance test. Thus, capacity planning of performance testing can be presented as follows:

Given a set of metrics, such as throughput and concurrent

users, which denote the target of workload generation over time T and workload model W, figuring out the number of machines that can satisfy a specific level of quality in terms of workload generation.

The following challenges are the principle challenges for capacity planning of performance testing:

Estimation of service demand: Service demand is a basic parameter for capacity planning models. There are two methods for obtaining service demand: direct measurement and statistical inference[24]. With direct measurement, codes are injected into source code or operating system to get the service demand directly. Statistical inference counts the throughput, response time and CPU utilization of requests, and then solves regression equations [34]. For performance testing, we may have to test different kinds of systems with different workloads. It will be too costly to evaluate the service demands of large systems. Besides, the instances of cloud are less stable than physical machines in LAN according to our observation, which will cause more uncertainty of measurements. We will give the evaluation in section V.

Development of accurate capacity model: The capacity planning can be solved by using Queuing Network [22], Queuing Petri Net [19] or regression methods [34]. However, these conventional models may not be able to accurately model resource virtualization and new resource schedule algorithms. The complicated architecture of cloud platform will potentially cause some difficulties of modeling.

Overcoming performance variation: The quality degradation of cloud services may happen during a long time testing. It would cause the deceleration of sending performance testing requests. This may render the results of performance testing incorrect. We can apply extra resources and rearrange tasks in run-time to ensure the quality of the testing. Performance exception and outlier detection can be useful to detect performance degradation [9].

Which level of resource utilization to be applied in a model: The upper bound of metrics that a type of machine can reach is usually determined by the bottleneck, such as network and CPU. Our preliminary experiments, performed on both EC2 standard instances and physical machines in LAN, have shown the similar phenomena as shown in Figure 2. The experiment settings, including system under test, hardware and database, are the same to that in section 5. We apply linear regression to predict the number of VUs that a machine can support under a certain workload model. The scatters of circles are the predicted CPU utilization with calculated service demand per virtual user with a specific scenario. The throughput of requests cannot be further increased by adding the number of virtual users. The scatters with stars are to validate whether our prediction is accurate. In Figure 2, the actual utilization of CPU will not further increase along with the number of virtual users, which also means the throughput will not further obviously increase when the number of virtual users reached to 40 in the graph. In one word, applying 100% of resource utilization in a model will introduce much error based on our experiments. Empirically, the result of applying 80% of resource utilization would be acceptable in many cases. Figure 2 is typical result generated with our LAN based experiments. The cloud experiments show the similar profile. Different EC2 instances reaches the bottleneck at different CPU utilization. Faster machine will reach bottleneck at a higher CPU utilization.

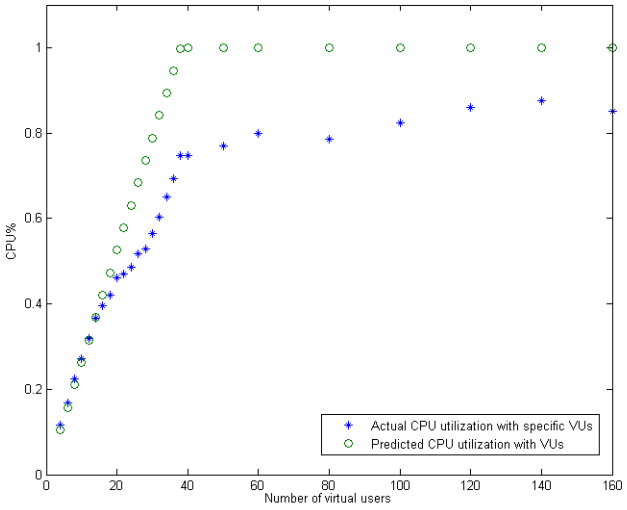


**Figure 2: The relationship between VU number and utilization of CPU.**

It is easy to understand that it is less possible for a test request be queued at a faster CPU.

### *4.1.3 Controlling of workload generation*

In goal-based performance testing, the workload distribution should be appropriate according to the objectives of workload characterization. However, there are potential run-time problems which may distort the arrangement of workload generation, such as bursty workload that saturate the test machines. Capacity planning and manual workload distribution are generally not adaptive, which is incapable of handling such run-time problems. For example, if a run-time error cause crash of workload generation, this would lead to the deviation of workload generation. This might eventually render the failure of a test. The run-time problems can be local or global (single or multiple machines). Both situations can potentially damage the quality of workload generation and accuracy of measurement.

The idea of autonomic performance testing might be helpful for controlling the quality of workload generation locally and globally. Adopting adaptive controllers on a test machine such as PID controller and Kalman Filter can dynamically adjust the workload generation locally. Global decisions can be further made by collecting feedback from each test machine to re-arrange test tasks or scale out. Besides, autonomic methods can complementary to capacity plan and manual workload distribution.

## 4.2 Data Analysis

The data analysis of performance testing is more complicated and harder than that in LAN. This is because of the instability and shared resources of cloud. Running the test scripts in clouds will introduce fluctuation of latency for each request [6], which brings challenges of analyzing test data for: 1) collecting test data, 2) data pre-processing, and 3) anomaly detection.

After execution of performance testing, testers need to collect and pre-process the required data for analysis. There are many problems make this step be challenging. One of the key problems is time synchronization, which can cause time

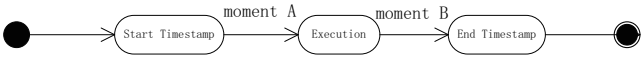


**Figure 3: Typical method of measuring an execution process**

stamp data to be inconsistent. For example, if we call "System.nanoTime()" in JAVA to record timestamps for some actions, it is hard to do synchronization for different CPUs. Because we don't know which CPU provides the services and it is also changing. The overhead introduce by hypervisor level scheduling will also cause measurement error as OS scheduling. Figure 3 shows a typical method of measuring an execution process. If a thread is switched out during the execution of code fragment (from moment A to moment B) that being measured, the measured execution time will be obvious enlarged. Thus, the existence of hypervisor level scheduling would increase the probability of such risks. Testers or cloud-based service providers need to look into such influences that may stain test data. Especially for large scale performance testing, synthetic workload is usually distributed to a limited number of test machines, if some of the machines are overloaded, then you would get some unexpected results.

Anomaly detection is usually performed to detect outliers in test data. Outliers can be the result of performance variation, service crash and so on. In order to correctly analyze the test results, we have to distinguish noise from outlier. This might be also more difficult for analyzing results of cloud-based tests. The measurement of request latency might be influenced by the noise introduced by instrumentation. When measuring fine-grained requests, the noise of measure potentially cause a measure to be an outlier. By controlling the quality of workload generation, this issue can be eased partly. However, one more level of scheduling potentially increases the risks of stained data.

## 4.3 Security

Most applications face the security problems when migrating to cloud. Balduzzi et al. have reported that both users and providers of public cloud may be vulnerable to security risks such as unauthorized access, malware infection, and loss of sensitive information [5]. Security problems of performance testing service have some traditional concerns, and some new issues on a cloud platform as well [27]. For performance testing, there are two key security concerns: 1) protection of critical information; 2) protection of performance testing services from illegal usage.

Protection of critical information is essential for service consumers. From the perspective of performance testing, the sensitive information can be critical information such as user names and passwords contained in test scripts or configuration files. It could lead to unimaginable damage if such information is leaked out. Besides, critical business information contained in scripts is also very confidential. The protection of critical information can be approached by two methods at different levels: access control and encryption. Access control can be implemented at both platform level and application level. Controls at platform level make sure users of cloud platform can only access their own cloud storage, and controls at application level make sure users of performance testing can only access the authorized data. Role-based access control can applied to solve such problems at different levels [29]. However, some information might be accidentally leaked out. In these cases, encryption is helpful to provide further protection of critical information. Symmetry key encryption can be used to encrypt sensitive data in scripts and configuration files [33]. Then, only the people that have the password can access the data without hacking. The implementation of the encryption should consider the overhead in performance testing. We should always keep in mind that introduce a new stuff should keep the precision of measured performance.

Protection of performance testing service from illegal purposes is also very important. This security concern exists because of heavy load that performance testing tool can generate with clouds. Attackers may be more willing to use cloud with a relatively cheaper price to implement an attack [27]. Service providers of performance testing should protect performance testing resources from being misused or used for illegal purpose. Therefore, service providers have to check whether the target system has authorized testers the right of conducting performance tests. The approach of LoadStorm [2] is to give tester a string of stochastic string which will be added to the code of web pages as a comment. During the test, the LoadStorm will check whether web pages include the authorized string. Such method is easy to implement, but it is fragile. There are at least two methods to bypass this mechanism. First is to apply man in the middle method to cheat the services when authenticating. Second is to create an own web site and set the authentication string in the web pages. Then, adding the reference of target system in the web pages. This will guide heavy load to the target system. Thus, authentication at page level is not sufficient at all. We are looking for a better solution that can achieve a good trade-off between the cost of authentication and the efficiency of testing.

The security requirements are different between the public cloud and private cloud. When on a private cloud, the resources and services are mainly provided for internal use. However, resources and services are assumed to be accessible for any people on public clouds. Attackers might be more likely to utilize public cloud for evil ideas because it is more difficult to obtain the access to a private cloud. The security levels and security concerns in different cloud systems are usually a trade-off among multiple factors.

## 4.4 Cost

The problem of cost is to figure out how much is needed for a performance test. It is one of the paramount concerns of consumers. They hope to apply the least resource with the minimum cost for a test. For example, a customer is going to do a load testing to reveal the throughput of a web site that might need more than 10000 virtual users to generate requests concurrently. He or she might raise a question that how much money is required for his task. If a customer buys performance testing services from service providers, it would be important for service providers to know the cost of their services for pricing. This problem can be abstracted to be costing model, which can be formalized as follows:

Assume the set of resources (hardware or software services) that charge for a specific rate to be:

$Rs = \{r_i | r_i \, is \, a \, resource \, that \, charges \, for \, a \, specific \, rate\}$

The corresponding rates of different resources to be:

$Rt = \{f(r_i, x_i) | r_i \in Rs\}$

where $x_i$ is a variable related with the rate, such as time, volume of storage, transferred data of network. For usage of CPU, time is usually used as the unit of rate. The costing model of a cloud platform can be simply denoted as:
$C = \sum_{i=1}^{n} \{f(r_i, x_i)\}$
The cloud service providers charge for resources like CPU, storage and network differently. The costing model should be capable of accurately predicting the cost of a test, based on which service providers can charge for a deposit from users. Such prediction models are usually piecewise linear multivariate functions, because a limited amount of usage may be free.

By analyzing the costing model of platforms, we can optimize the usage of resources to save the cost of performance testing accordingly. For example, Amazon does not charge any for all data transfer in EC2 and between Amazon EC2 and other Amazon web services within the same region (i.e. between Amazon EC2 US West and Amazon S3 in US West). We can take advantage of such free and cheap services.

There are some previous works about costing model, but not for cloud-based performance testing. In paper [10], the Amazon WAS based experiments showed that by provisioning the right amount of storage and compute resources, cost can be significantly reduced without significant impact on application performance. In paper [30], Singh et al. proposed a provision model using a multi-objective genetic algorithm formulation for performance-cost optimization in Grids, but their model focused on computing resources only.

## 4.5 Service level agreement

Service level agreement is a part of a service contract where the level of service is formally defined [3]. Service providers have to guarantee the service delivered to customers would satisfy the SLA items. If the quality of performance testing cannot be satisfied, the testing results will be useless and cause waste of money and resources. In other words, it is critical for customers to obtain reliable services of performance testing.

There are some challenges of achieving the goals of SLA, including problems of keeping, avoiding violating and verification of SLA. We discuss three main related problems as follows.

Performance variation problem: performance testing service providers usually buy IaaS services from companies like Amazon and Microsoft. These resources will be further shared by different customers. The quality of an application will be influenced by other applications on the same cloud [6][11]. The performance variation can also be caused by the consumers of performance testing services. The performance testing is strongly dependent on the quality of load generation. If the quality of service cannot be guaranteed, performance variation will cause the failure of test. Unfortunately, cloud service providers do not and may even not be willing to ensure the explicit performance guarantees for services in SLAs. It means the QoS of cloud and performance testing services cannot be guaranteed. Therefore, not all the performance testing scenarios are suitable to tested in cloud.Testing scenarios that need stable performance or even rigid requirements should have feasibility study before execution. Thus, performance variation awareness and follow-up strategies are needed to avoid violating the SLAs.

Resource allocation: The resources of a performance testing service provider may be limited. New performance testing tasks may influence the SLAs of existing tasks. We think that the service should be scheduled over committed requests automatically and provide friendly feedback to the customers. For urgent large tasks, new resources can be dynamically provisioned. Resource allocation and reservation mechanism can be used to ensure the quality of the service of performance testing and maximize the utilization of resources.

Trust: Consumers may not completely trust certain measurements provided solely by a single service provider and need to regularly employ third party mediators [26]. The precision of the measurement is a crucial issue, which may be the core focus of the trust problem. In order to solve the trust problem, a mechanism is required to enable third party measurement and assessment.

Patel, Ranabahu and Sheth proposed an architecture for managing cloud consumer and provider SLAs [26], based on WSLA specification[21]. Further researches could focus on the usage of SLA description language [20][25] and various measurement techniques of checking SLA levels [8] .

# 5. EVALUATION AND VALIDATION

Experimental evaluation and validation aims at addressing the issues above. We focus on the issue of workload generation quality on cloud. We use a famous E-Commerce demo JPetStore as a target system and producing synthetic workloads on both public cloud EC2 and LAN to produce a comparable results.

The server for the LAN experiments has a CPU of Intel Pentium dual-core E5400 and 4Gb memory. The operation system of the server is Windows server 2008. The client for generating the workload has a CPU of Intel Pentium dual-core E5700 and 2Gb memory. The test server for cloud experiments is EC2 standard extra large instance (API name: m1.xlarge). We evaluated three typical instance types as test clients: standard small instance (m1.small), standard medium instance (m1.medium) and standard large instance (m1.large). The benchmark for all experiments is JPetStore executed on Tomcat 7.0. The database is MySQL Community Server 5.5.27.

We use a simple workload model in our experiment as in Figure 4. It shows virtual users will do the three actions in "Login" session. Then virtual users will perform different sessions with different ratios. Finally, virtual users will do "signout". The think time is set to 3 seconds for all actions, thus we can adjust the load on a test machine by tuning the number of virtual users. The accuracy is calculated by: $(measured - set)/set$. The standard deviation and maximum deviation of think times are presented to show the stability of workload generation on different machines. Table 2 contains part of our experiments results which implies:

1. The generation of synthetic workload in LAN is more stable than cloud.
2. The stability decreases with the increase of load for both in LAN and cloud, but is sharper in cloud.
3. The stability and accuracy of load generation is related with the computing power of test machines. Faster machines generate better loads.
4. If the load for the test client is high, the test result of response time might be not accurate.

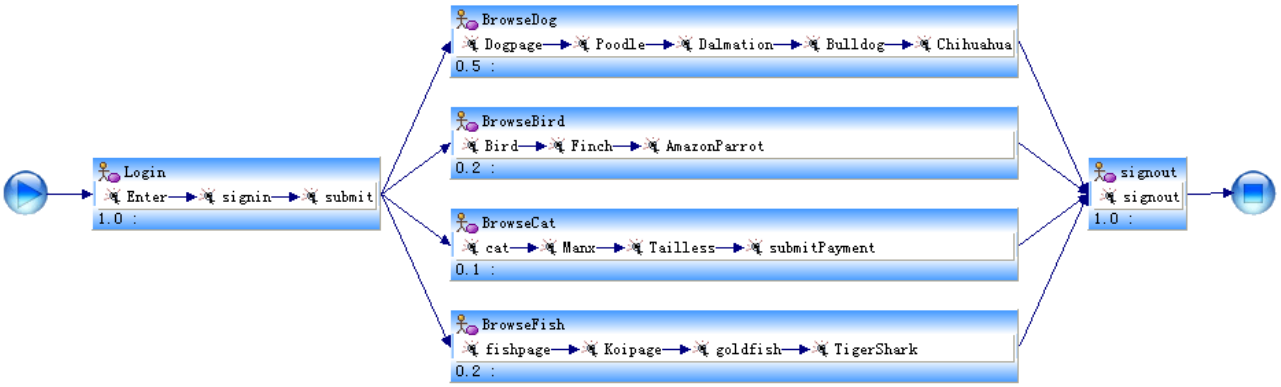


**Figure 4: A simple workload model used in evaluation**

**Table 2: Test results of evaluating the generation of constant think time with different numbers of virtual users**

| VUs | Think time measured | | Accuracy (%) | Std | Max Dev. (ms) |
|---|---|---|---|---|---|
| 50 | LAN | 3001.1 | 99.96 | 6.64 | 109 |
| | L | 3011.9 | 99.60 | 12.38 | 338 |
| | M | 3015.5 | 99.48 | 44.02 | 1836 |
| | S | 3018.8 | 99.38 | 38.57 | 511 |
| 100 | LAN | 3001.2 | 99.96 | 7.96 | 95 |
| | L | 3011.7 | 99.61 | 6.13 | 291 |
| | M | 3012.1 | 99.60 | 6.00 | 136 |
| | S | 3037.7 | 98.74 | 150.22 | 2648 |
| 200 | LAN | 3002.3 | 99.92 | 16.4 | 297 |
| | L | 3015.7 | 99.48 | 40.57 | 1181 |
| | M | 3017.4 | 99.42 | 61.44 | 1415 |
| | S | 3070.3 | 97.65 | 259.55 | 3975 |
| 250 | LAN | 3002.5 | 99.94 | 14.48 | 345 |
| | L | 3026.0 | 99.13 | 226.44 | 1369 |
| | M | 3018.7 | 99.38 | 39.43 | 666 |
| | S | 4465 | 51.17 | 6789.59 | 43683 |
| 500 | LAN | 3004.1 | 99.86 | 21.78 | 389 |
| | L | 3031 | 98.97 | 93.66 | 1213 |
| | M | 4251.3 | 58.29 | 2991.74 | 21090 |
| | S | 4503.7 | 49.88 | 3617.43 | 20589 |

# 6. CONCLUSION AND FUTURE WORK

In this paper, we visit the differences between conducting performance testing on LAN and cloud. We discuss relevant issues and challenges of cloud-based performance testing. When migrating performance testing to cloud, it is imperative to ensure the quality of workload generation. Our evaluation shows the existence of the issue of performance testing quality. Besides, the security problems may also hinder the spread of cloud-based performance testing. Other issues such as cost, data analysis and SLA also have many challenges of solving them. We hope the information mentioned above can help practitioners to avoid winding path when migrating performance testing to cloud. Not all the performance testing activities are suitable to be performed in the cloud. Before migration, the risks and benefits, what not to do and what not, should all be well analyzed.

We are applying feedback controllers to adaptively adjust the workload generation to improve the quality of workload generation locally. We will further integrate the global controlling to conduct the goal-based performance testing. We are also interested in how the virtualization, CPU bottleneck, I/O bottleneck and network delay influence the quality of synthetic workloads generation. The awareness of the measurement error at a lower level other than application level is promising to improve the accuracy of measurement.

# 7. REFERENCES


[1] De-facto standard for enterprise performance testing. http://www8.hp.com/us/en/software/software-product.html?compURI=tcm:245-937011&pageTitle=performance-center.

[2] LoadStorm. http://loadstorm.com/.

[3] SLA. http://en.wikipedia.org/wiki/Service-level_agreement.

[4] SOASTA. http://soasta.com/.

[5] M. Balduzzi, J. Zaddach, D. Balzarotti, E. Kirda, and S. Loureiro. A security analysis of amazon's elastic compute cloud service. *Proceedings of the 27th Annual ACM Symposium on Applied Computing - SAC '12*, page 1427, 2012.

[6] S. K. Barker and P. Shenoy. Empirical evaluation of latency-sensitive application performance in the cloud. In *Proceedings of the first annual ACM SIGMM conference on Multimedia systems - MMSys '10*, page 35, New York, New York, USA, 2010. ACM Press.

[7] G. Candea, S. Bucur, and C. Zamfir. Automated software testing as a service. In *Proceedings of the 1st ACM symposium on Cloud computing - SoCC '10*, page 155, New York, New York, USA, 2010. ACM Press.

[8] A. Chazalet. Service Level Checking in the Cloud Computing Context. In *2010 IEEE 3rd International Conference on Cloud Computing*, pages 297–304. IEEE, July 2010.

[9] L. Cherkasova, K. Ozonat, J. Symons, and E. Smirni. Anomaly? application change? or workload change? towards automated detection of application performance anomaly and change. In *2008 IEEE International Conference on Dependable Systems and Networks With FTCS and DCC (DSN)*, pages 452–461. IEEE, 2008.

[10] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good. The cost of doing science on the cloud: The Montage example. In *2008 SC - International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–12. IEEE, Nov. 2008.

[11] J. Dittrich and J.-a. Quian. Runtime Measurements in the Cloud : Observing , Analyzing , and Reducing Variance. *Proceedings of the VLDB Endowment*, 3(1), 2010.

[12] Poor application performance translates to lost revenue, research shows, 2008. http://www.networkworld.com.

[13] Z. Ganon and I. E. Zilbershtein. Cloud-based Performance Testing of Network Management Systems. In *2009 IEEE 14th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks*, volume 00, pages 1–6. IEEE, June 2009.

[14] 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. *2010 Third International Conference on Software Testing, Verification, and Validation Workshops*, pages 428–433, Apr. 2010.

[15] L. Hu and M. Xiao. The future of automatic test system (ATS) brought by Cloud Computing. In *2009 IEEE AUTOTESTCON*, pages 412–414. IEEE, Sept. 2009.

[16] W. Jun and F. Meng. Software Testing Based on Cloud Computing. *2011 International Conference on Internet Computing and Information Services*, pages 176–178, Sept. 2011.

[17] W. Kim, S. D. Kim, E. Lee, and S. Lee. Adoption issues for cloud computing. In *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia - MoMM '09*, page 2, New York, New York, USA, 2009. ACM Press.

[18] T. M. King and A. S. Ganti. Migrating Autonomic Self-Testing to the Cloud. In *2010 Third International Conference on Software Testing, Verification, and Validation Workshops*, number 1, pages 438–443. IEEE, Apr. 2010.

[19] S. Kounev. Performance Modeling and Evaluation of Distributed Component-Based Systems Using Queueing Petri Nets. *IEEE Transactions on Software Engineering*, 32(7):486–502, July 2006.

[20] D. Lamanna, J. Skene, and W. Emmerich. SLAng: A language for defining service level agreements. In *The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems, 2003. FTDCS 2003. Proceedings.*, volume 34069, pages 100–106. IEEE, 2003.

[21] R. Ludwig, H., Keller, A., Dan, A., King, R., Franck. Web service level agreement (WSLA) language specificatio. 2003.

[22] D. A. Menascé, V. A. F. Almeida, and L. W. Dowdy. *Performance by Design - Computer Capacity Planning by Example*. 2004.

[23] P. Mell and T. Grance. Draft nist working definition of cloud computing - v15. Technical report, 2009.

[24] G. Pacifici, W. Segmuller, M. Spreitzer, and a. Tantawi. CPU demand for web serving: Measurement analysis and dynamic estimation. *Performance Evaluation*, 65(6-7):531–553, June 2008.

[25] A. Paschke. RBSLA A declarative Rule-based Service Level Agreement Language based on RuleML. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 2, pages 308–314. IEEE, 2005.

[26] P. Patel, A. Ranabahu, and A. Sheth. Service Level Agreement in Cloud Computing. *Cloud Workshops at OOPSLA09*, pages 1–10, 2009.

[27] K. Popović and v. Hocenski. Cloud computing security issues and challenges. In *MIPRO, 2011 Proceedings of the 34th International Convention*, pages 344–349, 2010.

[28] L. M. Riungu, O. Taipale, and K. Smolander. Software Testing as an Online Service: Observations from Practice. *2010 Third International Conference on Software Testing, Verification, and Validation Workshops*, pages 418–423, Apr. 2010.

[29] R. Sandhu, D. Ferraiolo, and R. Kuhn. The NIST model for role-based access control. In *Proceedings of the fifth ACM workshop on Role-based access control - RBAC '00*, pages 47–63, New York, New York, USA, 2000. ACM Press.

[30] G. Singh, C. Kesselman, and E. Deelman. A provisioning model and its comparison with best-effort for performance-cost optimization in grids. In *Proceedings of the 16th international symposium on High performance distributed computing - HPDC '07* page 117, New York, New York, USA, 2007. ACM Press. ,

[31] N. Snellman, A. Ashraf, and I. Porres. Towards Automatic Performance and Scalability Testing of Rich Internet Applications in the Cloud. *2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications*, pages 161–169, Aug. 2011.

[32] Mission to Mars, 2012. http://www.soasta.com/performance-testing/mission-to-mars/.

[33] C.-C. L. S.-Y. Tseng;. Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter. In *Proceedings IEEE International Conference on Application- Specific Systems, Architectures, and Processors*, pages 277–285. IEEE Comput. Soc, 2002.

[34] Q. Zhang, L. Cherkasova, N. Mi, and E. Smirni. A regression-based analytic model for capacity planning of multi-tier applications. *Cluster Computing*, 11(3):197–211, Mar. 2008.