Experiences Virtualizing a Large-Scale Test Platform for Multimedia Applications

Robert Lübke
Computer Networks Group
Technische Universität
Dresden, Germany
robert.luebke@tu-dresden.de

Daniel Schuster
Computer Networks Group
Technische Universität
Dresden, Germany
daniel.schuster@tu-dresden.de

Alexander Schill
Computer Networks Group
Technische Universität
Dresden, Germany
alexander.schill@tu-dresden.de

ABSTRACT
Testing is an essential part of software development and many test platforms exist to facilitate the process. Test systems are scarce, because especially scalability tests require many computational resources. In this paper we show that these limitations can be overcome by migrating the test infrastructure into Cloud environments. Concrete virtualization concepts for large-scale testbeds are discussed using the example of NESSEE - an emulation environment for testing distributed Audio/Video conferencing applications. Furthermore, we describe how the Cloud migration allows us to better integrate the test runs of the platform into the work flow of software development.

Keywords
Cloud Migration, Software Test, Network Emulation, Continuous Integration, NESSEE

1. INTRODUCTION
Testing is of prime importance in today's process of software development. As system complexity keeps on growing, it gets more and more difficult to engineer faultless and accurate software products. As the testing process is mostly error-prone and time-consuming, test platforms and solutions are necessary for facilitation. Usually these testing platforms focus on only a few different goals of testing: functionality, stability, efficiency or scalability. Our use case of testing Audio/Video conferencing applications additionally requires network testing, as the functionality and end user experience heavily depend on the network conditions. These systems do not only require good networks with high bandwidths and low delay, but they also consume many computational resources due to real-time requirements.

Therefore, load tests and large-scale experiments with such systems particularly require many test systems. That is one reason why many universities and research institutions connected their test infrastructures with each other to build up large-scale testbeds for experimentation, for example PlanetLab [9] and GENI [1]. But on the other hand there are innovative companies which are usually not willing to join forces due to competition and for security reasons. As they do not use these open testbeds, they are bound to in-house experimentation solutions on dedicated test infrastructures. As test systems must be available at the desired point in time without any restrictions, typically large multitenant platforms are used. Even then there are sometimes situations in which multiple testers need to run large-scale tests concurrently and all available resources are used up. The typically employed solution to that problem is massive over-provisioning. One alternative to this cost-intensive approach is moving the test infrastructure into Cloud environments. These environments provide large numbers of compute and other necessary resources that can be used once they are required. Unneeded resources can be switched off to save energy and costs using Infrastructure as a Service (IaaS) solutions like Amazon Web Services[1] (AWS), Microsoft Windows Azure[2] and Google Compute Engine[3].

In this paper we present concrete concepts of how to virtualize a large-scale test platform for multimedia applications to be able to flexibly perform test runs in the Cloud. We also describe how Cloud migration can improve test automation. The testing platform that we use as an example in this paper is called NESSEE. It is a non-virtualized in-house emulation environment that is mainly used for experiments with Audio/Video conferencing systems with up to 1,000 participants. It covers functional, network but also scalability testing. As NESSEE requires a large number of test systems, it is an appropriate candidate for cloud migration. We further evaluate whether the high requirements of Audio/Video conferencing concerning computational and network resources can be met by the Cloud. In summary the main contributions of this paper are:

- general concepts of migrating a large-scale test platform for real-time multimedia applications to a Cloud environment,
- evaluation of the virtualization concepts regarding the gained scalability and accuracy of the test results.

[1]Amazon Web Services: `http://aws.amazon.com`
[2]Windows Azure: `http://azure.microsoft.com`
[3]Compute Engine: `http://cloud.google.com/compute`

TRIDENTCOM 2015, June 24-25, Vancouver, Canada
Copyright © 2015 ICST
DOI 10.4108/icst.tridentcom.2015.259664

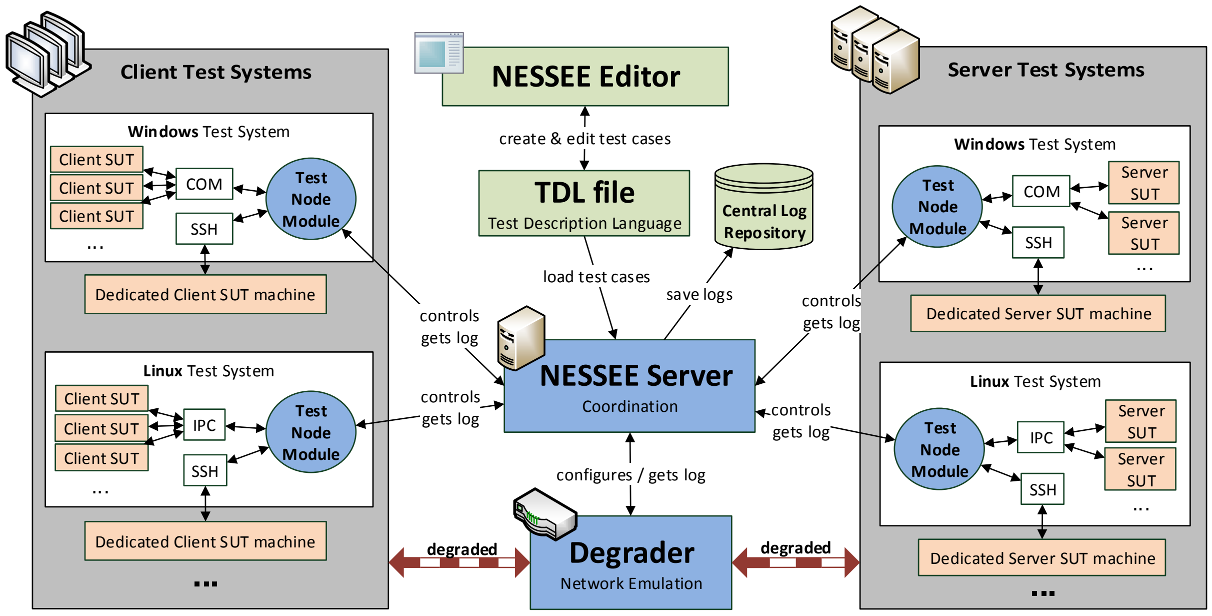

Figure 1: General architecture of the NESSEE platform (cf. [8])

In the remainder of this paper we discuss related research work and present the current status of the non-virtualized test platform NESSEE. As the current version of NESSEE has shortcomings concerning the scalability we propose concepts for Cloud migration in Section 4. After the evaluation of our concepts, we finally conclude the paper.

2. RELATED WORK

Virtualization is not a new approach in the field of networking testbeds, but it is actually used in systems like Federica [2], GLab [11] and Distem [10]. These approaches use virtualization on top of existing federated test beds. In-house test platform migration to public Clouds is not considered.

Public Cloud and IaaS solutions have intensively been used for large-scale testing in the research community before. The authors of [5] and [13] for example used the Amazon EC2 testbed to evaluate the scalability of their solution for global-scale PubSub communication and an XMPP-based infrastructure for the Internet of Things. These publications do only cover testing solution of specific systems, instead of generic test platforms featuring the reproduction of complex network structures.

There are also further research efforts about the general process of migrating existing IT systems to the Cloud. The authors of [3] present their experiences of migrating the service-oriented system *Hackystat* - a framework for facilitating the software development process. In [12] a general taxonomy and a cost model for Cloud migration of any application are described using a .NET application for Windows Azure as a case study.

To the best of our knowledge there is no published research effort about the special concepts and experiences for migrating a whole test platform for Audio/Video conferencing systems. Especially the existing research literature does not cover the reproduction of complex network structures and conditions using network emulation in Cloud environments.

3. STATUS QUO: THE NESSEE PLATFORM

The concepts, implementation details and evaluation results of NESSEE emulation platform have been published previously in [8] and [7]. This section should therefore only summarize the main concepts and general architecture of the emulation platform.

NESSEE mainly supports three kinds of test experiments. **Functional tests** can be performed on the API level of the Software Under Test (SUT). Various inputs can be passed to the SUT and the output is compared to the expected outcome. **Network tests** are essential for all applications that use network access, especially multimedia applications. The SUT is investigated under various network conditions like bandwidth limitation, delay, jitter, packet loss, reordering and duplication. **Scalability tests** are also supported. As the overall load is distributed over all test systems, multiple tests with over 1,000 concurrently emulated users of video conferencing applications have been performed before.

Figure 1 gives an overview about the **general architecture** of the emulation environment. As NESSEE mainly focuses on Client/Server-based distributed systems, we differentiate between *client- and server-side test systems*. These test systems run the so called *Test Node Module* (TNM) which is coordinated by the *NESSEE Server*. The TNM starts and controls multiple SUT instances on one test system to enable large-scale tests with limited resources. It uses a kind of inter process communication (IPC) to send calls to the SUT and evaluate return values and events. The Degrader is responsible for emulating the desired network conditions, that have been configured in a test case. As all traffic from the test systems is routed via the Degrader, the packet streams can be artificially worsened according to the test case configura-

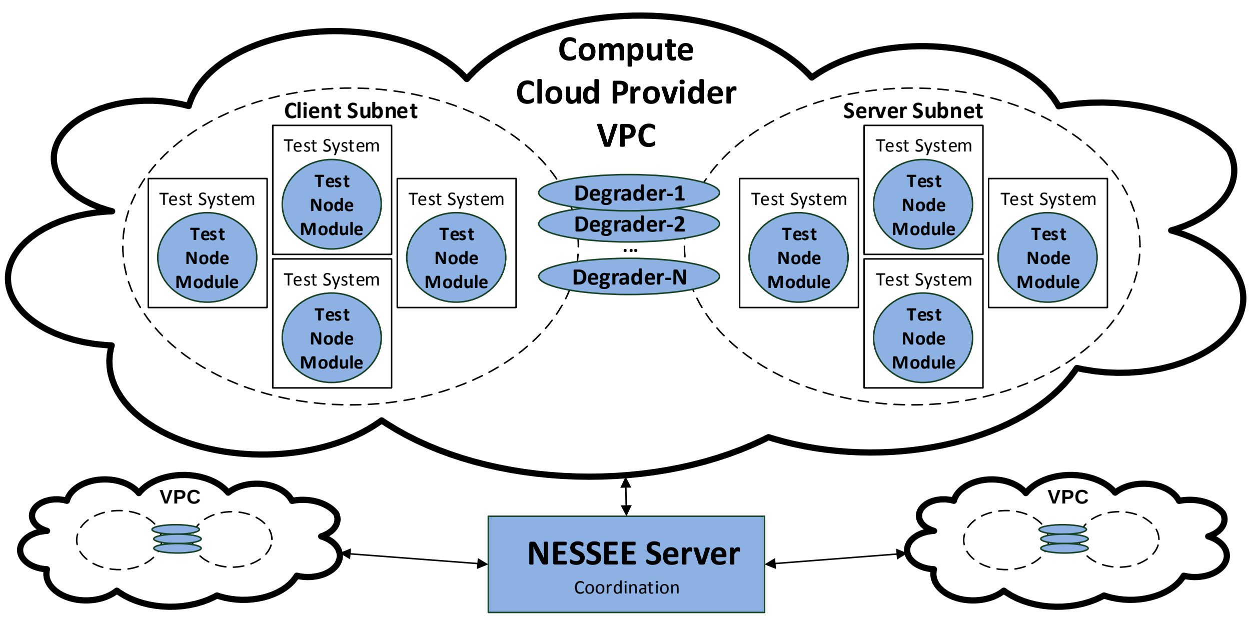

Figure 2: Proposed architecture of the test platform in a Cloud environment

tion. These test cases are defined in the generic, XML-based Test Description Language (TDL). TDL files can be edited manually or with the help of the graphical authoring tool *NESSEE Editor*.

Although the NESSEE platform is actually used regularly for large-scale tests of multimedia applications the single Degrader approach limits the scalability of the network emulation. The main bottleneck of the Degrader is neither the computing power nor the memory consumption, but the bandwidth of its network interfaces. Even multiple aggregated 10GBit interfaces are insufficient in large scenarios, if the configured bandwidth for the SUT should be guaranteed. Using multiple Degraders and dynamically changing the routing schemata of the fixed test infrastructure would only be a very laborious solution. Beside the general limitation of the number of test systems, large-scale network emulation is a problem that can be solved efficiently with the Cloud migration concepts that are described in the next section.

4. MOVING NESSEE TO THE CLOUD

There are two main reasons why to favor migration of the NESSEE platform or parts of it to a Cloud environment. First, the Cloud offers almost unlimited resources on demand, which allows us to run only the test systems that are actually required and to switch off all unneeded resources. Second, the Degrader scalability limitation can be overcome by performing network emulation on multiple virtual machines. This section discusses the concrete concepts addressing both aspects.

4.1 Adapted Architecture for Network Emulation in the Cloud

When migrating the NESSEE emulation environment to the Cloud, the general architecture is slightly adapted, which is also illustrated in Figure 2. All the test systems as well as the Degrader machines are located in a virtual network that is called Virtual Private Cloud (VPC). A VPC consists of two subnets, one for the server-side and one for the client-side test systems. The test systems themselves are virtual machines (VM) that are configured to be located in the corresponding subnet. Most Cloud providers offer fine-grained configuration of the VMs. According to the requirements one can set the number of virtual CPU cores and the sizes of memory and storage. The Degrader machines 1 to N shown in Figure 2 are VMs as well, but they are equipped with two network interfaces - one for each subnet. They are configured to route traffic from one subnet to the other. When a NESSEE test should be started, the required virtual machines are determined, launched and configured in the way that the traffic of the test systems is forwarded to the corresponding Degrader machines. These machines perform the network emulation as before using an own extension of the KauNet network emulator [4]. This emulator intercepts the packets and artificially worsens the network stream characteristic according to the configuration.

The proposed architecture allows to scale in different modes. One possibility is to start up the described setup for each NESSEE test run. Clearly, this has the advantage of concurrent NESSEE tests not influencing each other, but the Degrader machines will only have a very light load in most tests. Therefore, there is also the possibility to share network emulation resources among test runs as before. New virtual Degrader machines are only added to the setup, if a predefined load limit is exceeded. Figure 2 also shows that the central NESSEE Server can also control multiple VPC instances each containing multiple test systems. These VPCs are independent from each other and could also be hosted by different Cloud providers. In the end, it is also possible to virtualize the NESSEE Server itself as well, which enables the testers to easily set up a completely new and independent emulation platform.

4.2 Implementation

We chose to implement the Cloud migration concepts using the IaaS solution *Elastic Compute Cloud* (EC2) by Amazon. It offers a wide range of configurations and has different APIs and libraries for various technologies and programming languages. We further used a service called *Cloud Formation* that offers a template-based start-up and configuration. Using this approach one defines all required compute, storage, network and other resources in one place, for example a config file. When this configuration is uploaded to the Cloud provider, all resources can be started and stopped at the same time, which provides an easy way to manage and control a collection of related resources. We use *Cloud Formation* in the NESSEE platform for easily setting up the scenario shown in Figure 2 consisting of multiple compute instances, network interfaces, VPCs, subnets and the corresponding security settings.

4.3 Continuous Integration in the Cloud

The goal of the NESSEE emulation platform is not only to facilitate test runs and experiments, but also to enable as much automation as possible. The corresponding keyword in the context of software development is Continuous Integration (CI). CI helps to find errors as early as possible by monitoring changes in the code repository. If a programmer commits code changes, it triggers an automatic build of the project and runs the test routines. The corresponding developer is notified immediately upon occurring errors. Of course, it would be beneficial to not only run unit tests, but also integrate high-level NESSEE tests into the CI process. In the non-virtualized in-house test platform this is hard to realize because of many concurrent tests that need to run on a limited number of test systems. The CI process can therefore benefit from the proposed Cloud approach and the gain in scalability. Whenever a code change is detected in the repository the required resources can be launched for the test and shutdown afterwards.

Existing CI systems offer predefined tasks for running specific tests and parsing the results, but they also support web service calls and the execution of generic scripts. This mechanism is used to enable existing CI systems to make use of NESSEE tests. Therefore, the NESSEE Server provides different APIs for CI systems as well as Web and mobile apps. The NESSEE Server automatically deploys build results as SUT to the TNMs. The CI system then starts, monitors and analyzes the NESSEE test runs. The outcome of the test is then given to the CI system, which can notify the developers about possible malfunction.

In this way test platforms like NESSEE can be seamlessly integrated into the development cycle of the software under test. CI benefits from the combination with the proposed Cloud approach because even extensive use of testing is supported due to the almost unlimited resources of the Cloud.

5. EVALUATION

In this section we first evaluate the accuracy of the network emulation in the virtual environment compared to a setup with dedicated hardware. Secondly, we demonstrate the scalability of the Cloud approach by showcasing a large-scale test with its relevant characteristics.

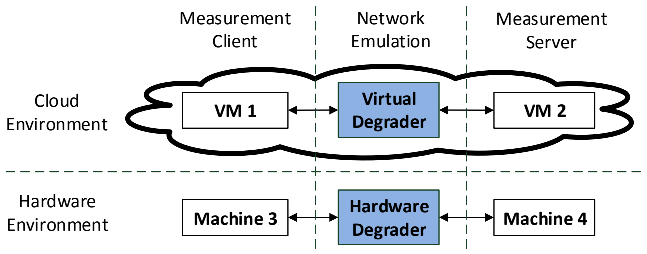

Figure 3: Environments for measuring the accuracy of the network emulation in the Cloud- and hardware-based approach

5.1 Accuracy of the network emulation

The measurement environment shown in Figure 3 is used to evaluate the accuracy of the network emulation. The Degraders each have two network interfaces with a measurement machine attached. We configure the Degraders to emulate the connection between both measurement nodes with different values for bandwidth (0.5, 7.2, 54, 61 and 160 MBit/s), delay (10, 50, 100, 500 and 1000 ms) and packet loss (0.01, 0.1, 1, 5 and 10 %). The tools `iperf`[4] and `ping` are then used to determine the actual values which can be compared to the expectation. Our own network measurement tool NORA [6] is used in addition to assure the quality of the results.

We use AWS again as Cloud Provider for the virtual environment. Amazon offers various instance types with different characteristics. In the measurement we use the following five virtual and physical machines as basis for the Degrader:

- `t2.micro`: low power with 1 virtual CPU core (vCPU), 1 GB RAM, low to moderate networking performance.
- `t2.medium`: like `t2.micro` but 4 vCPU and 4 GB RAM
- `c4.large`: compute optimized instance with 2 vCPU, 3.75 GB RAM, moderate networking performance
- `c4.8xlarge`: like `c4.large` but 36 vCPU, 60 GB RAM, 10 Gigabit network interfaces
- `Hardware`: Intel Xeon E5620 CPU @ 2.4GHz with 16 cores, 4 GB RAM, 2 interfaces with GBit Ethernet

The Measurement nodes *VM 1*, *VM 2*, *Machine 3* and *Machine 4* have better performance than the best Degrader instance to assure that they do not become a bottleneck during the measurement. Figure 4 shows the resulting deviations of configured bandwidth and loss values. The loss values are emulated with a very high accuracy on all machines. The results show only two outliers with an absolute deviation of ±0.1 % which still is acceptable. All other values vary from −0.05 to +0.03 % around the expected value. In comparison to the virtual Degraders, the hardware setup emulates more accurately. The largest deviations can actually be observed with the smaller virtual instance types. The results of the bandwidth measurements in Figure 4 are

[4]Iperf: `http://sourceforge.net/projects/iperf/`

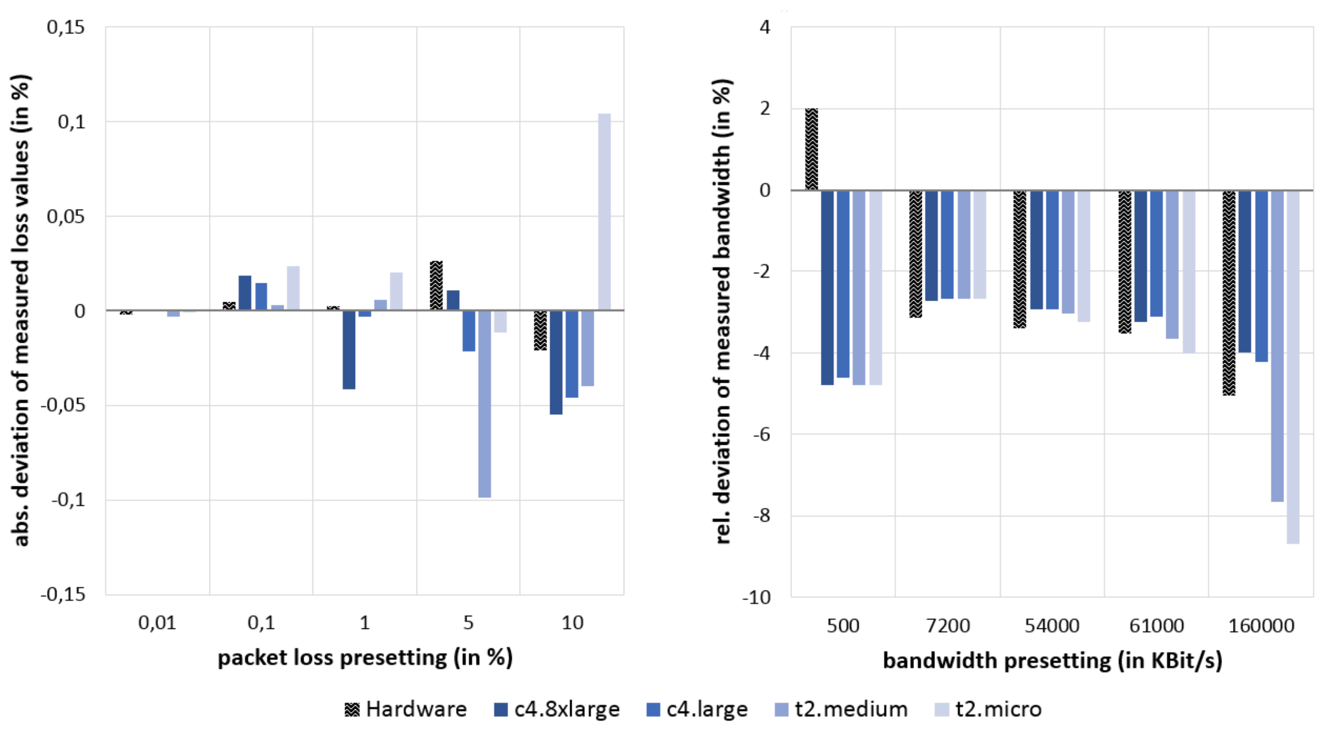

Figure 4: To evaluate the accuracy we determined the deviation of configured and actually measured network parameters in hardware and virtual environments with four different instance types.

very uniform. The relative deviations are mostly about −3 to −4 %, i.e. the measured bandwidths are mostly lower than the expectation. There are no significant differences in the accuracy results of the virtual and hardware-based environment. The mean relative deviation for all presettings is 3.42 % for the hardware Degrader and 3.53 % for the best virtual Degrader (using `c4.8xlarge`). The largest deviations of about 8 % can again be observed with the smaller instance types `t2.micro` and `t2.medium`. Regarding the delay measurements, we found that values of the hardware Degrader are very accurate. All values are only about 0.7 ms above the expectation, which is caused by the base delay of the measurement setup. Using the virtual Degraders, on the other hand, we discovered that the measured values are always 5 ms lower than the configuration. So, all packets are released too early to the network. Concerning delay emulation there are no differences in the accuracy emulation between the four used virtual instance types.

After performing the measurements and evaluating the results we can conclude that accurately emulating network conditions is also possible in Cloud environments using virtual compute instances. Although, we discovered significant differences in the accuracy when using different virtual instance types. The smaller instances with less computational power and networking performance consistently show the worst results. Though the hardware Degrader achieves the best results in all categories, the measurements of the two compute-optimized virtual instances only slightly differ. Finally, we can state that the benefit of scalable network emulation using the Cloud approach only comes with a small trade-off concerning the accuracy, if appropriate virtual machines are used. In our use case of large-scale testing of Audio/Video conferencing applications - and for sure in many other use cases as well - the required accuracy can be achieved in virtual Cloud environments as well.

5.2 Scalability

To even further demonstrate the feasibility of migrating the NESSEE test platform into the Cloud environment and to validate the scalability, we describe a large-scale test run, which we performed in both environments. The goal of the test is to investigate the functionality of a video conferencing server cluster under load condition with a conference consisting of 1,000 participants. The test setup consists of seven server test systems forming the cluster, one Degrader and 10 client test systems each running 100 SUT instances.

The client test systems in the hardware setup have an Intel Xeon CPU E5-2640 @ 2.5GHz with 8 Cores and 12GB RAM. We have chosen the `c4.2xlarge` with Intel Xeon E5-2666 v3 @ 2.9GHz and 15GB RAM as a corresponding virtual instance type, although its computational power is supposed to be better. Both types use Windows Server 2008 64bit as operating system.

In the performed test all SUT instances are started, logged in to the conference, and start receiving the videos of the other participants. Six participants actually start sending their video. The interface of the video server is used to check if every participant is present and receives or sends the video in the correct resolution and with the desired frame rate.

We found no significant differences between Cloud and hardware setup concerning the start-up time of the test. Starting and initializing 1,000 SUT instances took about 19 s in both environments. When all SUT instances are actually logged in to the conference, they are supposed to send and receive videos in 320x240 pixel resolution. In our experiments the correct resolution was always achieved for all emulated users. The participants in the hardware setup received the videos with about 26 frames per second (fps) and the Cloud setup achieved a mean value of the configured 30 fps. Furthermore, the average CPU load during the test is about 62 % in the hardware setup and 23 % in the Cloud environment. The different results are caused by the better computational power of the virtual machines.

Finally concluding these measurements we can state that the high requirements of Audio/Video conferencing applications can definitely be met by Cloud environments. Due to more powerful test systems the virtual environment even achieved better results. Amazon updates their physical hardware and offered instance types regularly. Despite this, the main benefit of the Cloud approach is that all test systems are shut down after the test while the hardware machines idle until they are required again.

6. CONCLUSION

Many inhouse testbeds that are regularly used by companies suffer from the problem of limited scalability. We argue the necessity of migration to Cloud environments to solve this problem. General concepts for virtualizing a large-scale test platform featuring the reproduction of complex network characteristics are proposed using the example of the NESSEE emulation environment that is used to test Audio/Video conferencing systems. The concepts are implemented and evaluated using AWS as a provider for IaaS. The evaluation results show significant differences of the network emulation accuracy for different virtual instance types. Even when using powerful compute-optimized virtual machines, there is a slight decrease in accuracy after migrating the test platform to the Cloud in comparison to the hardware setup. In our use case of large-scale tests of Audio/Video conferencing applications we decided to accept this decrease to benefit from improved scalability in the Cloud environment. Of course, other researcher need to face this trade-off in other use cases as well, but they can build on the concepts and benefit from the experiences we described.

Acknowledgment

This work results from the NESSEE project that has been financed by Citrix Systems, Online Services Division.

7. REFERENCES

[1] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar. Geni: A federated testbed for innovative network experiments. *The International Journal of Computer and Telecommunications Networking*, 2014.

[2] M. Campanella and F. Farina. The federica infrastructure and experience. *The International Journal of Computer and Telecommunications Networking*, 2014.

[3] M. Chauhan and M. Babar. Migrating service-oriented system to cloud computing: An experience report. In *IEEE International Conference on Cloud Computing (CLOUD)*, pages 404–411, July 2011.

[4] J. Garcia, P. Hurtig, and A. Brunstrom. KauNet: A Versatile and Flexible Emulation System. In *Proceedings of the 5th Swedish National Computer Networking Workshop (SNCNW 2008)*, Karlskrona, Sweden, April 2008.

[5] R. Hong, S. Shin, Y. Yoon, A. Laxmankatole, and H. Woo. Global-scale event dissemination on mobile social channeling platform. In *2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, pages 210–219. IEEE, 2014.

[6] R. Lübke, P. Büschel, D. Schuster, and A. Schill. Nora: An integrated network measurement tool for end-to-end connections. In *13th International Conference WWW/Internet*, Porto, Portugal, Oct. 2014.

[7] R. Lübke, R. Lungwitz, D. Schuster, and A. Schill. Large-scale tests of distributed systems with integrated emulation of advanced network behavior. *IADIS International Journal on WWW/Internet*, Vol. 10(2), 2012.

[8] R. Lübke, D. Schuster, and A. Schill. Nessee: An in-house test platform for large scale tests of multimedia applications including network behavior. In *9th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (Tridentcom)*, Guangzhou, China, 5 2014.

[9] L. Peterson, A. Bavier, M. E. Fiuczynski, and S. Muir. Experiences building planetlab. In *Proceedings of the 7th symposium on Operating systems design and implementation*, OSDI '06, pages 351–366, Berkeley, CA, USA, 2006. USENIX Association.

[10] L. Sarzyniec, T. Buchert, E. Jeanvoine, and L. Nussbaum. Design and Evaluation of a Virtual Experimental Environment for Distributed Systems. In *PDP2013 - 21st Euromicro International Conference on Parallel, Distributed and Network-Based Processing*, pages 172 – 179, Belfast, Royaume-Uni, Feb. 2013. IEEE.

[11] D. Schwerdel, B. Reuther, T. Zinner, P. Müller, and P. Tran-Gia. Future internet research and experimentation: The g-lab approach. *The International Journal of Computer and Telecommunications Networking*, 2014.

[12] V. Tran, J. Keung, A. Liu, and A. Fekete. Application migration to cloud: A taxonomy of critical factors. In *Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing (SECLOUD '11)*, pages 22–28, New York, NY, USA, 2011. ACM.

[13] H. Woo, H. Kim, K. Kim, and D. Kim. A large scale presence network for pervasive social computing. In *Proc. of the IEEE Conference on Pervasive Computing and Communications (PERCOM Workshops)*, pages 145–150, 2013.