Prudent Practices for Designing Virtual Desktop Experiments

Peiyu Liu, Wenzhi Chen, Zonghui Wang, Lirong Fu
College of Computer Science and Technology
Zhejiang University
Hangzhou, China
{liupeiyu, chenwz, zhwang, fulirong007}@zju.edu.cn

***Abstract*—Virtual desktop technology aims at accessing a remote desktop by endpoint hardware. Great attention has been increasingly paid to virtual desktop since it can increase the utilization of computing resources and provide more flexible accesses. However, researchers have not yet come up with a comprehensive set of rigorous standards of experimental design and implementation in this field. Therefore, it is difficult to conduct prudent experiments, which is correct, real, and transparent.**

In this paper, we assess the experimental evaluations of recently published papers on desktop virtualization. We observe that most works can be further improved, due to the unsuitable experimental environment and the lack of descriptions of experimental settings. In this paper, in order to help researchers, reviewers, and readers, we propose several guidelines for designing correct, real, and transparent desktop virtualization experiment.

***Index Terms*—Virtual Desktop; Prudent Experiment;**

I. Introduction

Nowadays, with the rapid development of virtualization technology, cloud computing has drawn great attention from both academia and industry fields [1], [2]. The thin client was first proposed in the 1990s, which finally develops rapidly in the form of virtual desktops in the boom of cloud computing [3], [4]. There are many mature solutions in the industry fields [5]. The academic community has also proposed a lot of impressive works [6]. Many desktop virtualization approaches will conduct extensive experiments to verify the performance of the proposed desktop virtualization systems. However, since there is no comprehensive set of rigorous standards of experimental design and implementation in this field, researchers will face numerous pitfalls during the experiment.

In this paper, we investigate issues about the prudent experimental evaluation of desktop virtualization systems. We observe that there are no general experimental standards in this field, which makes it difficult to compare these projects equally. In our previous research work, we notice that it is hard to reproduce the experiments that are conducted in many other approaches. Therefore, we put forward that the existing systems of desktop virtualization could be further improved in a rigor experimental standard. We solemnly declare that we highly respect the existing works. Under the purpose of helping researchers, we point out the current common problems that every relevant researcher, including ourselves, might encounter. One of these problems is that the description of the experiments in many works is sometimes inadequate. In addition, there are more serious issues that may affect the correctness of the experiments.

Our goal is to establish a set of rigor guidelines for the design, implementation, and description of prudent desktop virtualization experiments. We regard *correctness*, *reality*, and *transparency* as three cornerstones. Based on these cornerstones, we propose guidelines that can help researchers in prudent desktop virtualization experiments. We review existing 17 papers under our guidelines to confirm the validity of these guidelines. Most papers can benefit from our proposed guidelines. Through simple case study, we also validate the existence of some pitfalls encountered in the desktop virtualization experiments.

In summary, our work makes the following important contributions.

(1) We discover the common pitfalls in desktop virtualization experiments.
(2) We propose a set of guidelines to help researchers improve the rigor of desktop virtualization experiments.
(3) We investigate 17 desktop virtualization papers and validate that our guidelines are practical.
(4) We conduct simple case study to prove that the pitfalls in the desktop virtualization experiments deserve great attention.

II. Related Work

In this section, we summarize existing works related to our research.

Virtual Desktop Infrastructure (VDI): The technology that allows users to run desktop operating systems on virtual machines is known as VDI [6]. VDI manages a virtual desktop, which is a desktop environment in a virtual machine (VM) that runs on a centralized or remote server. A user can access a virtual desktop through a variety of terminals. There are many vendors which provide VDI solutions such as Citrix [5] and VMware [7].

Desktop as a Service (DaaS): DaaS provides the benefit of VDI without the cost and risk of managing physical resources, which allows a user to access desktop applications by any devices in anywhere [8]. Deboosere et al. propose a system

DOI reference number: 10.18293/SEKE2019-086

architecture to provide efficient desktop services in a cloud [9], which specifically focuses on mobile users. Therefore, due to resource constraints, virtual desktops are executed remotely. Kim et al. design and implement a desktop virtualization system using lightweight display protocols based on cloud DaaS [10].

Virtual Mobile Infrastructure (VMI): VMI extends the principles that allow VDI to run desktop applications on desktops and mobile devices - only this time, mobile apps are accessed remotely from mobile devices [11]. Su et al. propose vMobiDesk, a prototype system which provides mobile users with remote accesses to virtual mobile desktop such as Android desktop [12].

Prudent Practices for Designing Experiments: The rigor of experiments is very important for all academic papers. Previous researchers have analyzed the experimental rigor in other research fields [13]. However, before our paper, the rigor of desktop virtualization experiments has not been studied.

III. Guidelines for Prudent Experiment

We propose that the main pitfalls in existing desktop virtualization experiments can be divided into three categories. First of all, reasonable experimental setup and environment are the keys to ensure the *correctness* of the experiment. In addition, only real-world testings can *realistically* demonstrate that the virtual desktop really satisfies the actual needs of users. Finally, a *transparent* description of the experimental details can help the reviewers and readers understand the experimental setup and ensure the reproducibility of the experiment. Based on the above cornerstones, the following content outlines the guidelines of prudent virtual desktop experiments.

A. Correct Setting

*1) **Select the appropriate test indicators**:* The indicators tested in each desktop virtualization approach are not always the same. It is worthy to point out that the testing of parts of indicators is indispensable. For instance, we observe that many papers do not adequately test response time in virtual desktop experiments. However, for desktop virtualization, the performance of a system is strongly related to user interactions, no matter how appreciative other test results in the experiments are, the lack of response time testing will always be confusing. We emphasize that it is necessary to use appropriate methods to test important indicators.

*2) **Comprehensively consider the effects of experimental equipment**:* Various devices can be used in the tests of the performance of desktop virtualization systems. Obviously, the equipment used by the researchers is not uniform. It is indeed difficult to require everyone to use the same devices. We will not make such unreasonable demands. However, researchers should realize that different equipment may cause deviations in experimental results. We point out that different devices should be used in the same tests to decrease the experimental bias.

*3) **Determine the impact of network settings**:* Different network configurations are likely to significantly affect the performance of a virtual desktop. In many existing papers, the authors have configured excellent network connectivity, which is almost impossible in real life. In such a network environment, desktop virtualization systems may perform very well. However, virtual desktops may not perform so well in daily network environments. We propose to configure a variety of network environments for comparison.

*4) **Pay attention to the effects of desktop resolutions**:* In desktop virtualization systems, desktop data can be encoded with any resolution when it is transmitted. When the same desktop data is transmitted by different resolutions, the amount of the data is different, which will have an impact on the network bandwidth and response time. Therefore, we emphasize that authors must pay attention to the influence of resolutions and evaluate them in their papers.

B. Realistic Tests

*1) **Conduct real-world experiments**:* Desktop virtualization works need to solve real-life problems, which should be able to work in the real world while satisfying the needs of users. To evaluate the actual performance of a desktop virtualization system, real-world experiments should be conducted. We propose that using the equipment of users to conduct experiments in the real-world working environment is more convincing.

*2) **Be cautious about the compatibility**:* For various reasons, many papers are evaluated only in a single Operating System (OS) version. However, considering the compatibility, we propose that papers should explain whether the current desktop virtualization system can be applied to other OS versions, or investigate in detail how much work is needed to port the desktop virtualization system to other OS versions.

C. Transparent Description

*1) **Detail description of the OS and tools used in the experiment**:* Different OS versions and test tools may lead to different experimental results. We insist that the author is obliged to elaborate the OS version and test tools in their paper, such as "Windows 7 64-bit none third-party programs installed", "netperf-2.7.0 released on 21 Jul 2015, download address: https://github.com/HewlettPackard/netperf/releases", The detail description can improve the possibility of reproducing experiments by readers.

*2) **Explain the reasons for the poor/outstanding performance**:* If the virtual desktop does not perform well on a test, we strongly recommend that the author should analyze the possible causes of the poor performance carefully. It is always a respectable practice to propose possible improvement solutions. Even if the desktop virtualization system performs well in the experiments, the author still needs to perform a comprehensive analysis. If the experimental environment, such as the condition of the network, is the reason for the outstanding performance of a desktop virtualization system, then ignore this reason is unfair for other papers.

TABLE I
LIST OF SURVEYED PAPERS CLASSED BY TOPIC. SOME TITLES ARE SHORTEN WITH [...].

#	Authors	Title	Venue
VDI			
1	Baratto et al. [14]	MobiDesk: Mobile Virtual Desktop Computing	ACM MobiCom 2004
2	Baratto et al. [6]	THINC: A Virtual Display Architecture for Thin-Client Computing	ACM SOSP 2005
3	Kibe et al. [15]	The Evaluations of Desktop as a Service in an Educational Cloud	IEEE NBiS 2012
4	Alexander et al. [16]	Building a Cloud Based Systems Lab	ACM SIGITE 2012
5	Darabont et al. [17]	Performance Analysis of Remote Desktop Virtualization based [...]	MACRo 2015
6	Kim et al. [10]	Cloud-based Virtual Desktop Service Using Lightweight [...]	IEEE ICOIN 2016
7	Uehara et al. [18]	Performance Evaluations of LXC based Educational Cloud in a [...]	IEEE WAINA 2017
8	Triyason et al. [19]	The impact of screen size toward QoE of cloud-based virtual desktop	Elsevier PCOCEDIA 2017
DaaS			
9	Beaty et al. [8]	Desktop to Cloud Transformation Planning	IEEE ISPDC 2009
10	Cristofaro et al. [20]	Virtual Distro Dispatcher: a light-weight Desktop-as-a-Service [...]	Springer CLOUD 2009
11	Lai et al. [21]	A Service Based Lightweight Desktop Virtualization System	IEEE ICSS 2010
12	Calyam et al. [22]	Utility-directed resource allocation in virtual desktop clouds	Elsevier COMNET 2011
13	Deboosere et al. [9]	Cloud-based Desktop Services for Thin Clients	IEEE INTERNET COMPUT 2012
VMI			
14	Hung et al. [11]	Executing mobile applications on the cloud: Framework and issues	Elsevier COMPUT MATH APPL 2012
15	Nguyen et al. [23]	An Efficient Video Hooking in Androidx86 to Reduce Server [...]	Springer CUTE 2014
16	Su et al. [12]	vMobiDesk: Desktop Virtualization for Mobile Operating System	IEEE HPCC 2017
17	Wang et al. [24]	FUSION: A Unified Application Model for Virtual Mobile [...]	IEEE DSC 2017

TABLE II
CRITERIONS FROM THE CORRESPONDING GUIDELINES. '$\star\star\star$': MUST COMPLY. '$\star\star$': SHOULD BE FOLLOWED. '$\star$': GOOD TO MEET.

Criterions	Guidelines	Rating	Explication
Correct Setting			
Response time test	2.1.(a)	$\star\star\star$	Perform response time test in a suitable manner
Bandwidth evaluation	2.1.(a)	$\star\star\star$	Test bandwidth in an appropriate way
Frame rate test	2.1.(a)	$\star\star\star$	Test the frame rate in a reasonable way
Different devices	2.1.(b)	$\star\star$	Use different client devices for multiple sets of tests
Diverse network connectivity	2.1.(c)	$\star\star\star$	Configure a variety of network environments for comparison
Different resolutions	2.1.(d)	$\star\star\star$	Test virtual desktop performance under different resolutions
Realistic Test			
real-world experiments	2.2.(a)	$\star\star$	Conduct experiments in the real users' work environment
Real users	2.2.(a)	$\star\star\star$	Ask real users to experience the system and measured users' ratings.
Multiple OSes	2.2.(b)	$\star$	Conduct experiment with different server OS versions
Transparent Description			
Introduction of OS version	2.3.(a)	$\star\star\star$	Detaile the OS version
Introduction of test tools	2.3.(a)	$\star\star\star$	Describe the selected test tool in detail
Interpretation of poor performance	2.3.(b)	$\star\star\star$	Analyze the possible causes of the poor performance carefully
Analysis of good performance	2.3.(b)	$\star\star\star$	Conduct in-depth analysis of good performance
Improvement solutions	2.3.(b)	$\star$	Propose possible improvement solutions

IV. ASSESSMENT OF GUIDELINES

In this section, we elaborate the assessment of the guidelines presented in previous sections. The assessment method we use is to extract criterions from the guidelines and then apply the criterions to 17 recent papers listed in Table I for analysis. Through this method, we validate the practical value of the guidelines and obtain some observations.

A. *Process of Assessment*

In order to assess our guidelines, we extract more specific criterions from the corresponding guidelines. Table II shows the criterions we proposed. For researchers who study on desktop virtualization, every criterion we define can be judged directly with the corresponding papers. As shown in Table II, we divide all the criterions into three levels. Among them, level '$\star\star\star$' indicates that the experiments in a rigorous paper

TABLE III
OVERVIEW AND BRIEF DESCRIPTION OF OBSERVATIONS.

Criterions	Rating	Yes	Weak	Description
Correct Setting				
Response time test	⋆⋆⋆	8 (47.0%)	2 (11.8%)	Only about half of the papers performed response time test in a suitable manner. In addition, there are two papers that measured response time, but their conclusions are too sloppy. None of the remaining articles measured response time.
Bandwidth evaluation	⋆⋆⋆	9 (52.9%)	0 (0%)	About half of the papers do not measure bandwidth consumption at all.
Frame rate test	⋆⋆⋆	3 (17.6%)	0 (0%)	Almost all papers have not evaluated the FPS of virtual desktops.
Diverse network connectivity	⋆⋆⋆	5 (29.4%)	0 (0%)	Only five papers configured a variety of network environments for comparison.
Different resolutions	⋆⋆⋆	6 (35.3%)	0 (0%)	Less than half of papers tested the performance of virtual desktops at different resolutions.
Different devices	⋆⋆	2 (11.8%)	0 (0%)	Most papers do not mention support for multiple client devices. In this case, client compatibility cannot be evaluated.
Realistic Test				
Real users	⋆⋆⋆	2 (11.8%)	0 (0%)	Few researchers have invited real users to participate in the evaluation of desktop virtualization systems. In other words, basically only the rigid numerical results are provided.
real-world experiments	⋆⋆	3 (17.6%)	1 (5.9%)	A majority of papers do not conducted real-world experiment. Only two papers met this criterion. In addition, there was a paper that used tools to simulate different user environments.
Multiple OSes	⋆	1 (5.9%)	0 (0%)	In only one case the authors conducted experiment with different server OS versions.
Transparent Description				
Introduction of OS version	⋆⋆⋆	10 (1%)	0 (0%)	Seven papers do not describe the operating system version of the virtual desktop. This puts those who wish to reproduce the experiment into a situation where they have no rules to follow.
Introduction of test tools	⋆⋆⋆	6 (35.3%)	2 (11.8%)	About half of the papers do not present information about the tool. There are also two papers that just mention the name of the tool but do not detail the version and other information.
Interpretation of poor performance	⋆⋆⋆	6(35.3%)	1(5.9%)	Only about a third of the papers explain the reasons for the poor performance in the experiment. There is also a paper that only explains a part of the poor performance.
Analysis of good performance	⋆⋆⋆	3 (17.6%)	2 (11.8%)	The vast majority of the papers just put some numerical results that look very good but do not explain any of the deep reasons behind the excellent results. There are also two papers that only explains a part of the good performance.
Improvement solutions	⋆	1 (5.9%)	0 (0%)	Only one paper proposes improved solutions after discussing the causes of poor performance.

must comply with these criterions. Level '⋆⋆' shows that these criterions should be followed, while Level '⋆' means that it is good to satisfy these criterions.

We leverage each criteria in Table II to evaluate the papers in Table I. Two of our authors conduct a survey for all the papers. Our goal is to investigate the prudence of the experiments through all the available information in the paper. Therefore, in the process of investigating the papers, we follow the rules that only focus on the content of the papers. We do not review the source code or contact the authors of the papers for more details. We use these restrictions because they actually reflect what readers and reviewers face. Reviewers are often not likely to investigate details that the author has omitted. In the case of a double-blind submission, there is simply no way to contact the author. In other words, only the paper itself can be easily accessed by reviewers and readers. It is the author's obligation to clarify the details in the paper.

B. *Observations*

Table III lists the statistics results of all the surveyed papers. *Yes* refers to papers adhere to the guideline.

1) Correct Setting: About 47% of the papers conduct experiments to test response time. Two papers take how users feel as a performance measure, which is not suitable. Similarly, half of the papers have bandwidth evaluation. For the frame rate test, more than 80% of the approaches ignore the test. No more than 50% of the papers use different devices to evaluate the proposed systems. Only 27.7% of the papers test different network connectivities. 30% of the works conduct the experiment of different resolutions. We can see that most papers do not include enough correct settings.

2) Realistic Test: The survey indicates that few papers conduct real-world experiments. From the table, we observe that 80% of the papers lack real-world experiments and only one

TABLE IV
SETTINGS FOR OUR EXPERIMENTAL PLATFORM.

Server Host	Processor	3.40GHz Intel Core i7-6700
	RAM	8 GB
	OS	Ubuntu 16.04
	Kernel	Linux 4.4.0
Server Guest	Virtualization tool	VirtualBox 5.2.10
	RAM	1 GB of RAM
	OS	Android-x86-5.1.1
Client Device	Hardware	Google Nexus 9
	OS	Android-6.0
	RAM	2GB
Network	Hardware	NETGEAR R8000
	Frequency	2.4GHz/5GHz
	Speed	1000MB

paper conduct experiments with different OS versions. Few researchers (about 11%) have invited real users to participate in the evaluation of desktop virtualization systems.

3) Transparent Description: Half of the papers do not introduce the OS version that they leverage. 50.5% of the papers do not describe the testing tools. Only half of the papers mention network connectivity. Consequently, in the majority of the cases, readers fail to figure out the experiment setup adequately, nor can repeat the experiments. Meanwhile, we find that more than 90% of the papers incompletely describe experimental results. Only two papers offer improvement solutions to address low-performance problems.

V. CASE STUDY

In order to explain the problems caused by the violation of the guidelines intuitively, we design experiments to prove that the aforementioned pitfalls will affect the evaluation results of a desktop virtualization system.

Specifically, we analyze two experiments related to two criterions: (1) the influence of resolution on experimental results, and (2) the impact of network connectivity. The experiments of the remaining criterions will be updated to *arc.zju.edu.cn*. Based on our previous experience in desktop virtualization, we believe that all the factors in the guidelines will affect the experimental results. The desktop virtualization system used in our experiment is vMobiDesk, which is a relatively new VMI framework [12].

A. Experimental Setup

As shown in Table IV, all tests are performed on a server machine with a 3.40GHz Intel Core i7-6700 processor and 8 GB RAM. The server machine runs Ubuntu 16.04 with Linux 4.4.0 kernel. The guest OS is Android-x86-5.1.1, which runs in a virtual machine created by VirtualBox. The tested client device is Google Nexus 9 running on Android-6.0 OS. A 100 Mbps, 1 ms latency LAN network is utilized to construct local 2.4GHz and 5GHz Wifi communication network between the mobile device and the server.

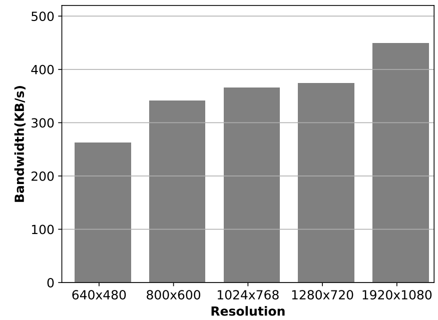

Fig. 1. Bandwidth consumption with different resolutions.

B. The Influence of Resolution

We measure the bandwidth consumption with different resolutions during browsing web pages by vMobiDesk with *iftop-0.17* (released on 12 Feb 2006, download address For http://www.ex-parrot.com/~pdw/iftop/download/iftop-0.17.tar.gz). This experiment uses a 2.4GHz wireless network. The results are shown in Table 1. The higher the resolution, the more data of the virtual desktop system needs to be transmitted. Therefore, the higher the bandwidth consumption is required. When the resolution is 640 x 480, the bandwidth consumption is only 262 KB/s. In contrast, it increases to 449 KB/s when the resolution is 1920 x 1080. The increment is up to 71.4%. Without proper handling, this huge consumption will obviously affect the accuracy of an experiment. For instance, a paper ultimately concludes that the desktop virtualization system takes up quite low bandwidth without claiming the use of 640 x 480 resolution for the experiment, which will lead to misunderstandings since the users still consider that the bandwidth consumption remains so small even with a higher resolution.

C. The Impact of Network Connectivity

We further measure the response time under different network connectivity. We conduct experiments in the wireless network environments of 2.4 GHz and 5 GHz, respectively. The average response time is evaluated by several operations such as opening an application, typing several words in a document and returning back to the home screen, etc.

One obvious result is that the response time under 5GHz Wifi is shorter than that under 2.4GHz Wifi. Regardless of the resolution, this difference caused by the frequency of the wireless network always exists. Taking resolution 1024 x 768 as an example, the response time under 5GHz wifi is 410 ms. However, the response time under 2.4GHz wifi rises up to 680 ms. Although this result is not as shocking as the previous experiments shown, 30% of time increment is also large enough to affect the accuracy of an experiment. Leveraging a 5GHz network in an experiment without elaboration will lead to a misunderstanding of the excellent performance.

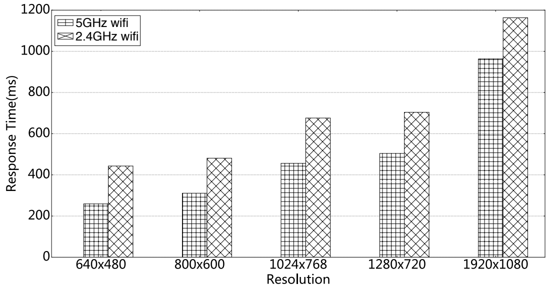

Fig. 2. Response time under different network connectivities.

VI. Discussion

As shown in Section V, improper design of experiments, ambiguous experimental descriptions, and other pitfalls will cause deviations in the assessment of virtual desktop and even lead to misunderstanding. We insist that it is the author's responsibility to ensure the *correctness*, *reality*, and *transparency* of their papers. Surprisingly and disappointingly, our research shows that considering rigorousness, the majority of experiments in desktop virtualization papers are subject to improvement. To reiterate, we greatly respect the researchers, including the authors of the papers we surveyed, and the results of their work. However, it turns out that we have not yet come up with a comprehensive set of rigorous standards of experimental design and implementation in the desktop virtualization field. We believe that a reasonable set of guidelines will help everyone greatly improve their efficiency of work and the quality of the paper. We hope our paper can bring some help to reviewers, authors, and readers. This is the initial motivation for us to carry out this work, and it is also our ultimate goal.

VII. Conclusion

In this paper, We summarize pitfalls that are often encountered when conducting desktop virtualization experiments. Based on this, we further propose guidelines that help researchers design and implement prudent experiments in desktop virtualization systems. We extract specific criterions from the corresponding guidelines and leverage each of the guideline criteria to evaluate the papers we selected. Our survey results validate that many papers can be improved. Using our proposed guidelines will help improve the quality of the papers in the experimental part. Finally, we conduct experiments and demonstrate the impact of some pitfalls succinctly.

VIII. Acknowledgement

We appreciate the anonymous reviewers for their helpful suggestions. Our paper is partly based on the work supported by the National Key R&D Program of China (No. 2016YFB0800201).

References

[1] B. Hayes, "Cloud computing," *Communications of the Acm*, vol. 51, no. 7, pp. 9–11, 2008.

[2] P. K. Sahoo, C. K. Dehury, and B. Veeravalli, "Lvrm: On the design of efficient link based virtual resource management algorithm for cloud platforms," *IEEE Transactions on Parallel and Distributed Systems*, vol. PP, no. 99, pp. 1–1, 2018.

[3] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper, "Virtual network computing," *IEEE Internet Computing*, vol. 2, no. 1, pp. 33–38, Jan. 1998.

[4] S. Haripriya., R. Indumathi., and M. S. Manikandan, "Virtual network connection using mobile phones," *Compusoft International Journal of Advanced Computer Technology*, vol. 3, no. 6, 2014.

[5] "HDX Technologies Optimize User Experience - Citrix," https://www.citrix.com/products/xenapp-xendesktop/hdx-technologies.html.

[6] R. A. Baratto, L. N. Kim, and J. Nieh, "THINC: A Virtual Display Architecture for Thin-client Computing," in *Proceedings of the Twentieth ACM Symposium on Operating Systems Principles*, ser. SOSP '05. ACM, 2005, pp. 277–290.

[7] "Horizon 7 | Virtual Desktop Infrastructure | VDI | VMware," https://www.vmware.com/products/horizon.html.

[8] K. Beaty, A. Kochut, and H. Shaikh, "Desktop to cloud transformation planning," in *IEEE International Symposium on Parallel and distributed Processing*, 2009, pp. 1–8.

[9] L. Deboosere, B. Vankeirsbilck, P. Simoens, F. D. Turck, B. Dhoedt, and P. Demeester, "Cloud-based desktop services for thin clients," *IEEE Internet Computing*, vol. 16, no. 6, pp. 60–67, 2012.

[10] S. Kim, J. Choi, S. Kim, and H. Kim, "Cloud-based virtual desktop service using lightweight network display protocol," in *International Conference on Information NETWORKING*, 2016, pp. 244–248.

[11] S. H. Hung, C. S. Shih, J. P. Shieh, C. P. Lee, and Y. H. Huang, "Executing mobile applications on the cloud: Framework and issues," *Computers and Mathematics with Applications*, vol. 63, no. 2, pp. 573–587, 2012.

[12] K. Su, P. Jiang, Z. Wang, and W. Chen, "vmobidesk: Desktop virtualization for mobile operating system," in *IEEE International Conference on High PERFORMANCE Computing and Communications; IEEE International Conference on Smart City; IEEE International Conference on Data Science and Systems*, 2017, pp. 945–950.

[13] C. Rossow, C. J. Dietrich, C. Grier, C. Kreibich , V. Paxson , N. Pohlmann, H. Bos, and M. V. Steen, "Prudent practices for designing malware experiments: Status quo and outlook," in *Security and Privacy*, 2012.

[14] R. A. Baratto, S. Potter, G. Su, and J. Nieh, "Mobidesk:mobile virtual desktop computing," in *International Conference on Mobile Computing and NETWORKING*, 2004, pp. 1–15.

[15] S. Kibe, T. Koyama, and M. Uehara, "The evaluations of desktop as a service in an educational cloud," in *International Conference on Network-Based Information Systems*, 2012, pp. 621–626.

[16] J. Alexander, A. Dick, J. Hacker, D. Hicks, and M. Stockman, "Building a cloud based systems lab," in *Conference on Information Technology Education*, 2012, pp. 151–154.

[17] Ã. Darabont, K. J. Kiss, and J. Domokos, "Performance analysis of remote desktop virtualization based on hyper-v versus remote desktop services," *Macro*, vol. 1, no. 1, pp. 125–134, 2015.

[18] M. Uehara, "Performance evaluations of lxc based educational cloud in a bare metal server," in *International Conference on Advanced Information NETWORKING and Applications Workshops*, 2017, pp. 415–420.

[19] T. Triyason, W. Krathu, T. Triyason, and W. Krathu, "The impact of screen size toward qoe of cloud-based virtual desktop," *Procedia Computer Science*, vol. 111, pp. 203–208, 2017.

[20] S. Cristofaro, F. Bertini, D. Lamanna, and R. Baldoni, "Virtual distro dispatcher: A light-weight desktop-as-a-service solution," in *International Conference on Cloud Computing*, 2009, pp. 247–260.

[21] G. Lai, H. Song, and X. Lin, "A service based lightweight desktop virtualization system," in *International Conference on Service Sciences*, 2010, pp. 277–282.

[22] P. Calyam, R. Patali, A. Berryman, A. M. Lai, and R. Ramnath, "Utility-directed resource allocation in virtual desktop clouds," *Computer Networks*, vol. 55, no. 18, pp. 4112–4130, 2011.

[23] T. D. Nguyen, C. T. Huynh, H. W. Lee, and E. N. Huh, "An efficient video hooking in androidx86 to reduce server overhead in virtual desktop infrastructure," in *Ubiquitous Information Technologies and Applications*. Springer Berlin Heidelberg, 2014, pp. 107–114.

[24] C.-M. Wang, Y.-S. Wu, and H.-H. Chung, "Fusion: A unified application model for virtual mobile infrastructure," in *IEEE Conference on Dependable and Secure Computing*, 2017.