Muhammad Moiz Arif
EV 3.274, 1515 Saint-Catherine Street West
Department of Computer Science and Software Engineering, Concordia University
Montreal, QC, Canada H3G 2W1

June 11, 2017

Dear Editor and Reviewers:

Thank you for your insightful feedback and comments, both positive and constructive, and for allowing us the opportunity to improve our manuscript. We have taken each of your comments into consideration and made the appropriate changes and extensions to our manuscript.

Editor Comments

Overall the reviewers believe that the presented work is of great interest to the EMSE readers. All three reviewers are supportive of the publication of the work provided that the authors perform a major revision of the manuscript.

Comment E.1. *Reviewer 1 wants to see supporting evidence that VMs are widely used for performance testing. The reviewer points out there has been much work on performance testing – the reviewer points out related work – Nistor ICSE 13, Jin PLDI 12 that does not rely on VMs. Reviewer 3 tells you about the worrysome tendency that he sees in your paper to explain what it is that you are not doing, before explaining what you are doing. Please correct it in the revised version of the paper.*

Response. We have now added motivation to our paper by giving examples of online discussions and references of commercial software systems. Please see our response to comment R1.1. We have also added discussions to the related work section on performance detection and bugs. Furthermore, we have rephrased and have used a direct approach in our explanations.

Comment E.2. *The authors should clarify points in the paper. Reviewer 1 states that the the paper is not clear about how the three aspects of testing results can help to find performance problems. Also, the reviewer needs clarifications for the second analysis - the correlation between metrics - why is it useful? Reviewer 2 points out that your claim "To the best of our knowledge, the discrepancy between performance testing results in virtual and physical environments has never been studied.", is arbitrary and does not hold. Reviewer 3 also complains that the test environments are not quite clear and that the paper needs clear and precise research questions.*

Response. We have clarified that our paper was set in the context of how performance testing results are analyzed in practice (the three aspects). We explained in detail how each aspect is useful to find performance problems and how our findings impact the use of each type of analysis. Please see the details to our responses to comment R1.3 and R2.29. Additionally, we have clarified an ambiguity present between our work and the related work. A description of testing environments has now been added.

Comment E.3. *Reviewer 3 Unclear research methodology, since it is not clear exactly what methodology is being followed. Moreover, the reviewer complains that the paper lacks actionable findings. The paper repeats in several places that there is performance of virtual and physical environments. That is interesting, but it is not actionable. The authors should work more on the paper to determine how their solution can contribute to the practice by offering actionable items based on your findings.*

Response. We improved our description to research methodology. We also have rephrased the implications and actionable findings of the paper. We not only emphasize that developers should not add overhead from virtual environments as is, but we also say that in order to normalize the performance metrics between the

two environments prior approaches may not work. We propose an approach to minimize the deviance hence developers should leverage such an approach before processing performance testing results. We also discuss the impact of our findings on each type of analysis on performance testing results. Please see our responses to comments R2.29 and R3.43.

Below, we include a description of the changes that we made to our manuscript with respect to each of the reviewer's comments. We denote the reviewers comments in italic typeface, and our responses follow below each reviewer comment.

Reviewer 1 Comment

Comment R1.1. *Is there any evidence supporting that VMs are widely used for performance testing? There has been much work on performance testing (e.g., Nistor ICSE 13, Jin PLDI 12) that does not rely on VMs. The paper should provide more details on the background and motivation for this study, such as why and in what circumstances VMs are used in performance testing.*

Response. Thank you for pointing this out. The motivation of this paper started from our extensive collaboration with our industrial practitioners, where we found that a large amount of the performance testing is conducted in virtual environments. In order to confirm that our experience with the industry is not an exceptional case, we find online discussions and posts by developers and testers about performance testing in virtual environment [5][10][23]. In addition, we find that the developers of SugarCRM are documenting their experiences in conducting load testing (which has a big overlap with performance testing) on virtual environments [15].
Based on these cases, we are confident that our experience, with our industrial practitioners who rely on virtual environments for performance testing, is not an exceptional case. In fact, Reviewer 3 also supports the fact that using virtual environments in performance testing is common. We add this in the introduction section.

Comment R1.2. *The study does not address the most important problem in performance testing, i.e., fault detection. Even if the discrepancy exist, thzere is no evidence showing that such discrepancy can affect testing effectiveness. The ultimate objective of performance testing is to find performance bugs. It would be more convincing if the authors can evaluate the discrepancy in finding bugs between VMs and physical environments.*

Response. We totally agree with this comment. The ultimate goal is to investigate the impact on the detection of real world performance bugs. This paper is rather the first step to lay a ground to deeply understand such discrepancy. Without such knowledge, direct evaluation of performance bugs would be like treating the effect than the cause. With the knowledge of such discrepancy, we can better, in the future, understand the existence and magnitude of impact on detecting real world performance bugs. In addition, another contribution of our paper, besides the road towards real world performance bugs, is the effort of identifying approaches that may minimize the discrepancy. In particular, we find that the approach proposed by Nguyen et al.[16] for workloads in heterogeneous environments does not effectively reduce the discrepancy and we propose a new approach that is shown to be more effective. Our future work on evaluating the impact of real world performance bugs will be based on the testing results that are processed with the reduced discrepancy. We add more discussion according to this comment in our conclusion.

Comment R1.3. *The paper is not clear about how the three aspects of testing results can help to find performance problems. Especially for the second analysis - the correlation between metrics - why is it useful?*

Response. The investigations in this work are based on the following 3 types of analyses:

- The first type of analysis to identify the trends and shape of the distributions of performance metrics. Due to the difference between testing environments, performance testing results are expected to be different in raw value. However, the shape of distribution and the trend should be similar. For example, when there is a higher load, CPU increases. If in one environment, we observe the CPU has increasing trend while not seeing the same trend in another environment, we observe a discrepancy. Therefore, we use Q-Q plots and normalized KS tests to examine the differences in trends and shape of the distributions.
- The second type of analysis, is to identify the combination between two performance metrics. As claimed by Cohen et al. [4], combinations of performance metrics are significantly more predictive towards performance issues than individual metrics. We believe that a change in these combinations of relationships can reflect the discrepancy of performance in the two environments.
- The third type of analysis is used to see the combination of all performance metrics all together by constructing a statistical model. Similar approach has been adopted by Xiong et al. [25, 4]. This analysis also serves as the baseline for our future study i.e. fault detection.

We have addressed and rephrased this in the new revision.

In short, the three types of analyses are three ways that performance engineers may exploit to examine the performance testing results. We would like to set our study in such a context and see if the differences between virtual and physical environment would impact such analyses, instead of simply checking the raw value of performance metrics, which is expected to always have differences.

Comment R1.4. *In the related work section, instead of just describing the three types of analysis, the authors should relate the existing work to the proposed study. Does the discussed existing work rely on VMs? If it does, are there any problems caused by the discrepancy between VMs and physical environments?*

Response. Prior research does not explicitly mentions the environment. Other prior research is performed on either virtual or physical environments only. For example, vPerfGuard [25] is only conducted in the virtual environment. However, none of them discuss the issue with being on both virtual and physical environments. This motivates our research. We added such discussion in our related work section to clarify.

Comment R5.5. *As virtualization becomes wildly adopted, many companies use virtual environment as their production environment to reduce operation costs. Does that invalidate the purpose of this study? What if an application is actually deployed in a virtual environment?*

Response. Thanks for the comment. We agree that some of the applications may later be deployed in a similar virtual environment. However, there still exist many scenarios that would value our study. As a first scenario, many systems have historical performance testing results that are based on the physical environments [3]. When comparing the new performance testing results and the old performance testing results, such discrepancy needs to be known. Second, software systems are often released both in virtual environment as cloud based services or on-premise. For example, SugarCRM [22] and BlackBerry Enterprise Service [1] sell their systems both on-premise and on cloud based services. Ensuring the consistency between two solutions, or knowing the need (or to what extent) repeating performance tests in different environment is important to save resources. In addition, we also evaluate the discrepancy between performance tests both on virtual environment and with different virtual machines, which is also valuable to performance testing and system deployment that are carried in virtual environments. We clarify this and add more discussion about this in the introduction of our new revision.

Comment R1.6. *Section 3.3: "the workload of the performance tests is varied periodically in order to avoid bias from a consistent workload" - how did it get varied.*

Response. We varied it randomly by the number of threads and ensuring that the variation was identical between both the environments. We elaborate such detailed information in our new revision.

Comment R1.7. *"The work-load variation was introduced by the number of threads." Why not consider other types of workloads, such as the amount of input data? Increasing the number of threads may also be used to speedup the performance.*

Response. Thanks for the comment. We agree that there exist many ways of varying workload. We opt to use the similar performance tests as prior studies that analyze performance testing results [17], since we want to set our study in that context. Using other types of variations is planned for our future work. We discuss this limitation and the plan of future work in our new revision of the paper.

Comment R1.8. *The quality of the performance tests could greatly influence the testing results. The paper should provide more details. Examples of performance tests can be useful. Also, how many performance tests are used in the study? Why does a test take so long (9 hours) to execute? Is running performance tests twice sufficient to reduce influence of randomness?*

Response. We apologize for the confusion. We answer the questions accordingly:

1. We use the same performance tests as used in the related studies [8][13][17][6].
2. Performance tests are typically run for a long period of time, similar to prior studies [8][13][17][6]. The long length of performance tests reduces the risks that are introduced due to unstableness of the system, the missing coverage of certain load and the lack of statistically significant data for analysis. In future work, we can extend our study to run for even a longer period of time (e.g., 72 hours).
3. We do not guarantee that running it twice would completely eliminate the randomness. However, the consistency between the two runs provides evidence that the randomness of each run would not have significant impact on the validity of our results. Our discussion on the variance between the same tests in the same environment is included in the discussion section.

   We have clarified all of the aforementioned points in the updated version of our manuscript. In addition, we discuss the quality of performance tests as a possible threat to validity to our findings.

Comment R1.9. *Different performance tests may performance different functionalities (e.g., SQL query VS server restart). The functionalities should be evaluated separately.*

Response. We decided to test the subject system as a whole. Although testing a smaller unit of functionality may benefit in locating performance issues. However, the goal of the paper is to examine the discrepancy of the system as a whole, which is closer to the performance impact on real users. In our future work, we will separate different components of the system and conduct lower level, isolated performance tests, in order to investigate the impact on performance bugs. We add the discussion in the new revision of our paper.

Comment R1.10. *Section 3.1: what is the size of each application?*

Response. The size of each application is added in the new revision.

Comment R1.11. *On page 7, the design choice of combining metrics of two datasets is not justified.*

Response. Similar to comment R1.9, we treat the system as a whole. The discussion of this choice is added

into the new revision.

Comment R1.12. *On page 7, realistically, interference on the real-world systems cannot be restricted like the one mentioned in the setup. The concern is that, by leaving out the system load, the statistical model might miss the opportunity to adjust to the real-world situation. Also, different assumptions about the system workload could affect the choice of the statistical model and thus perturb the prediction results.*

Response. We agree that in real world, the systems may have different interference to impact their performance. However, in our experiments, we opt for a more controlled environment to better understand the differences without any interference, hence we can limit the chance that the discrepancy is from handling interference rather than the environments. Future work can be applied to investigate the performance impact from different environments by handling interference, with having the knowledge of environment discrepancy. On the other hand, we agree that system load counters may illustrate valuable knowledge of the system. We include throughput metrics that are associated with system load. In addition, we will include more metrics in our future work. We discuss the above points in our new manuscript threats section.

Comment R1.13. *On page 15, what is the purpose of removing "metric that has a higher average correlation with all other metrics"?*

Response. We used such an approach to remove multicollinearity that is present between performance metrics. In fact, the approach is based on a popular statistical analysis [11]. In essence, whenever two metrics are found to be highly correlated, one of them needs to be removed. To determine which one to remove, the approach chooses the one that is more highly correlated with the rest of the metrics. We elaborate the description of our approach in the new revision of the paper.

Comment R1.14. *On page 16, what regression model is used? On page 17, linear regression model is mentioned briefly. Why is a linear model chosen? Not until on page 19, the assumption of a linear relationship is mentioned. A brief writing of the design decision would be more appropriate.*

Response. We chose linear regression model as it is used in prior work [25][20]. More importantly, the linear model is more straightforward to explain compared to that other model. Hence, it is easier to interpret the discrepancy that are illustrated by the model. We discuss the decision in our revised manuscript.

Comment R1.15. *R-squared is used without explanation. If 10-fold cross validation has an explanation, R2 may deserve one too.*

Response. We provide this explanation in the revised manuscript.

Comment R1.16. *On page 18, "good model fit (66.9% to 94.6%)", is that the absolute percentage error?*

Response. Sorry for the confusion, the values are the R^2 of our models. We elaborate our text to avoid such confusion in the new revision.

Comment R1.17. *There is much work on performance testing and bug detection, which should be discussed in related work.*

Response. We include the new discussion in more related work about performance testing and bug detection. The list of newly added related work include:

- Nistor et al.: Toddler: Detecting Performance Problems via Similar Memory-access Patterns in ICSE '13 [19].
- Jin et al.: Understanding and Detecting Real-world Performance Bugs in PLDI '12 [9].
- Nistor et al.: Discovering, reporting, and fixing performance bugs in MSR '13 [18].
- Tsakiltsidis et al.: On Automatic Detection of Performance Bugs in ISSREW '16 [24].
- Malik et al.: Automatic Comparison of Load Tests to Support the Performance Analysis of Large Enterprise Systems in CSMR '10 [12].
- Zaman et al.: A qualitative study on performance bug in MSR '12 [26].

Comment R1.18. *If the trace data can be made public, others may use it to replicate the experiments.*

Response. Thanks for the suggestion. We have now made the data public.

Reviewer 2 Comments

Comment R2.19. *This paper is championing the fact that there is a lack of compatibility or similarity between the performance metrics obtained from virtualized and physical environment. The purpose is secondary; whether the performance metrics collected are the solely the result of performance testing (load, stress, smoke, tortures or capacity testing) aka active testing OR collected during passive testing, i.e., field monitoring/testing for various purposes, i.e., anomaly detection, continuous validation of workloads, for future long and short term forecasts.*

Therefore, the claim "To the best of our knowledge, the discrepancy between performance testing results in virtual and physical environments has never been studied.", is arbitrary and does not hold.

Some work exists in that have implicitly compared the difference between the metrics harvested from the VMs and Physical machines, however, performance testing was not their main focus. A few explicit research efforts exist to gauge the difference between metrics collected from physical and virtual environment. I am pointing to one such work conducted by Netto from PUCRS and Sadd from Dell "Evaluating load generation in virtualized environments for software performance testing". They conducted several load tests on both virtual machines and physical machines.

Response. Thank you for your feedback.

We agree that there exist prior studies such as the one mentioned above that examine performance counters from both environments. However, the comparison is rather simple. For example, in the above mentioned study, the authors directly compare the metric values. In fact, difference exists between any two tests, even from the same environments, yet statistically, they may not. Therefore, we examine the performance testing results in the context that when performance engineers use different statistical analyses on these results and see whether the difference between VM and physical would impact the statistical analyses results.

We discuss the above difference between our work and prior studies in related work section. In addition, we have revised our manuscript to say our work is one of the first works that examine the discrepancy between performance testing results in virtual and physical environments.

Comment R2.20. *However, the use of VMs may introduce extra overhead (e.g., a higher than expected memory utilization) to the testing environment and lead to unrealistic performance testing results...Why testing over virtualized lead to unrealistic PT results?*

Response. Prior studies on systems and hardwares have investigated the overhead of virtual machines. For example, Huber et al. try to build a performance model to predict performance of applications that are migrated from a native system to a virtual environment or from a virtual environment to a new one [7]. There are similar studies which indicate that the overhead in virtual environment may hamper the results of performance tests [2][14]. However, it is found that such overhead cannot be simply added up to the existing system load, while it instead brings confounding effect on the system performance overall. Having known the existence of such overhead, interpreting performance testing results is not straightforward.

Comment R2.21. *Our findings show that practitioners cannot assume that their performance tests that are observed on one environment will necessarily apply to another environment... Is there any evidence to back up the current practice(s) in which practitioner generalize the result of one environment to another (phy to Vir), especially of the large scale software, you mentioned in the paper?*

Response. Thanks for the question. Such a question is also raised by reviewer 1. Please refer to our responses to comment R1.1 and R1.5. In short, we find that the use of virtual environment in performance testing is not an exceptional cases but rather common in practice. In addition, there exist scenarios that both physical and virtual environments are used for systems (e.g., services provided both on cloud and on-premise).

Comment R2.22. *Exploring, identifying and minimizing such discrepancy will help practitioners and researchers understand and leverage performance testing results from virtual and physical environments... Why would they *LEVERAGE* performance testing results from virtual and physical and to achieve *WHAT* purpose. Bench-marking folks won't like this idea*

Counter narrative: Many companies also try to virtualize their load generation infrastructure which seems like a good idea for maintenance and elasticity reasons. However, they defiantly do tests on physical environment for the show-off purpose of their performance. This is a fact that you can squeeze the most performance out of physical environment. Nevertheless, especially for SaaS infrastructure, or to cater the need of stakeholders and large clients, virtualized environment are used and performance results obtained are attributed to specific virtual environment. That's why the benchmark teams exists in large enterprise that catalog the performance under different virtualized environment. Application of findings from one environment to another is not a usual practice .

Response. Thanks for the question. Such a question is also raised by reviewer 1. Please refer to our responses to comment R1.5. We also discuss the impact of our findings on the analyses of performance testing results.

Comment R2.23. *Since the authors have framed the paper as an empirical study, it is very important to provide the necessary information in the paper for replication purpose. Replication, allows other researcher to validate authors claim(s) and in cases, compare it with their own techniques/cases and findings. What are workload parameters are used in this paper for DS2 and for Cloud Store using JMeter?*

Response. We will share our data and our load driver configuration for replication purposes.

Comment R2.24. *Did you used the same hardware for setting up the virtual environment?*

Response. Yes, we have used the same hardware for setting up the virtual environment. We clarify this in the new revision.

Comment R2.25. *Among the three machines, on which machine you were running perfmon agent (remote collection)? OR you were running it on all three machines?*

Response. We use perfmon to monitor on the two machines that run web(app) server and the database server. We do not monitor performance of the machine but instead the performance of the web application process and the database process directly, in order to minimize the influence of the perfmon. We clarify this in the new revision.

Comment R2.26. *How did you ensured that environment remained constant for each performance test? For example after few tests, the disk may get full hence IOPS can get impacted.*

Response. We restore the environments and restart the systems before conducting every test. We add such clarification in the new revision.

Comment R2.27. *For both the Q-Q plots for D2 and Cloud store, the metrics do not have the same trend. Did you used the samples of the performance tests results when the test was in equilibrium? If yes how did you ensured that?*

Response. Yes, we waited till the point of stability before processing. We achieve this by not including data from the beginning and ending of a test (10 minutes each end). This was a consistent practices across all our tests.

Comment R2.28. *Also, before calculating the correlation among the performance metrics from virtual and physical environments, did you removed ramp-up and ramp-down observations of the performance metrics for a test? The system is usually not stable during warm-up and cool-down period. Ramp-up and down periods, for a test repeated multiple time under a given workload, and constant environment, many not necessary be correlated to each other.*

Response. We completely agree with the comment and the sample was taken after the warm up, the ramp-up and before the cool-down period. We ran the warm up period for 2 minutes. We remove the recorded performance metrics during this time frame. As explained in comment R2.27, we removed the recorded data during our ramp-up and cool down period in order to ensure equilibrium.

Comment R2.29. *What is the implication of this study?*

Response. There exist three actionable implication from this study:

- Developers cannot assume an straightforward overhead from the virtual environment, (such as a simple increase of CPU).
- Prior approach that are proposed to normalize performance testing results with different loads may not work between physical and virtual environments.
- We propose an approach that may minimize the discrepancy between performance testing results from virtual and physical environments. Developers should leverage such an approach before processing performance testing results from virtual and physical environments.

We highlight the actionable implications at the end of each research question of in in our new revision. In addition, in our new revision, for every research question, we add a paragraph to discuss the impact of our results on practitioners interpretation of performance testing results using each type of analysis.

Comment R2.30. *In order to assist the practitioners leverage performance testing results in both environments, we also investigate ways to transform results from virtual and physical environments and performance metrics based on deviance may reduced [REDUCE] the discrepancy between performance metrics... Weird sentence, unless am reading between lines.*

Comment R2.31. *...such challenges, virtual environments (i.e., VMs) are often leveraged for performance testing [8,8,47]... why "8" is repeated in the ref?*

Comment R2.32. *...ents, such overhead would not significantly impact on the practitioners who examine the performance testing results...significantly impact WHAT on the practitioners?*

Comment R2.33. *...paper, we perform a study on two open-source systems, DS2 [13] and CloudStore [10], where performance tests are conducted on[USING] virtual and physical environments*

Comment R2.34. *we study whether the performance metric follow[S] the same shape of [THE] distribution and the same trend in virtual and physical environments.*

Comment R2.35. *...which can lead to [A] different set of conclusions.*

Comment R2.36. *For example, [THE] virtual environment has a CPU's utilization spike at a certain time [,] but the spike is absent in the physical*

Comment R2.37. *...there exist[S] a plethora of VM software*

Comment R2.38. *...high when we normalize by [THE] load as per Equation*

Comment R2.39. *In the reference section, there is an extra '24' hanging at the end of the page*

Response. All above typos are fixed, except for comment R2.39, which is not a typo but the page number. We also proofread it again to eliminate language issues.

Comment R2.40. *Overall, interesting research and paper was a joyful read.*

Response. Thank you!

Reviewer 3 Comments

Comment R3.41. *The test environments are not quite clear. This concern is minor because it could be fixed with complete descriptions of the hardware and virtual environment configuration. For example, is the physical server disk setup a RAID? SSD? SATA/NVMe? Is the network traffic generated locally or is it simulated from another machine (which would be influenced by the network hardware)? Perhaps more importantly, what is the disk setup on the virtual machine? One of the findings in the paper is that the I/O metrics differ more than the CPU/memory metrics when compared to the physical machine. That is definitely not surprising if the VM is configured to use VDI or another virtual disk, given the overhead in mimicking a drive within an existing filesystem. It is possible to use disk passthrough in Virtual Box and other VM software. In that case, a physical disk is passed by the host OS directly to the guest OS. That setup tends to be much much closer to native speed.*

Response. Thank you for your feedback in helping us make it a stronger manuscript. We have now provided a detailed description of environment configuration in the new draft. The network traffic was generated on another machine. As we were well below the capacity the network overhead did not make a difference. In addition, as the network setup was same for both of our environments the effect would cancel out during the analysis. One of the concern was about using disk passthrough. We opted to not use disk passthrough mainly due to the fact that disk passthrough was not enabled in our experiences of collaborating with our industrial partners, which primarily motivates this work. The main reason is the portability issues, since practitioners want to quickly deploy an existing vm image that's designed for performance testing and start performance tests right away, there are other discussions online about why disk passthrough may not be a good idea[21]. Based on such information and mainly practices in our industrial collaboration, we opted not to enable it. We elaborate our choice in the new revision.

Comment R3.42. *Unclear research methodology. At several points, the paper is not clear on exactly what methodology is being followed. One potential deal breaker for this paper is that the test scenarios for the virtual and physical environments are "similar" so the distributions should have the same shape (first sentence on page 10). The test scenarios should be identical, not just similar. Similar implies that the authors created a test scenario for each environment using the same tools. They should instead have created just one test scenario. If the test scenarios are not the same, then the discrepancy could obviously be due to a different scenario. One way to ensure that the tests are the same is to record the network traffic of one run and use the recorded traffic as input for the next run. In a virtual environment, one can even use the same time and date settings.*

The paper needs clear and precise research questions. The paper jumps into methodology and analysis without clearly explaining what is to be analyzed. For example consider section 4.1, "Examining individual performance metrics." It is not explained exactly what an "individual" metric is. I had to try to understand it from reading the Approach subsection in 4.1. The concept is not that difficult; the section is just comparing, e.g., a CPU metric in one environment with the same metric in the other environment. It just needs to be explained more clearly up front. Some of the language is confusing, such as "intuitively the scales of performance metrics are not the same." I am not sure if the sentence refers to scales not being the same across metrics or across environments.

The authors have tendency to explain what they are not doing, before explaining what they are doing. That is confusing because it makes it difficult to understand the methodology. For example in Section 4, "We do not predict... Instead, our experiments are set in..."

All of these issues with unclear research methodology could be fixed by adding clear research questions and a methodology for answering each question. The bigger problem here is that the methodology is unclear enough that it might be obscuring more important problems under the surface.

Response. Thanks for your suggestion, we improve our explanation of the empirical study in our new draft to avoid confusions by having 3 research questions. We also rephrase the texts to improve the ease of understanding our approaches.

In particular, we did not adopt an approach that monitor the network traffic to control load. Our experiments are trying to replicate the practice of our industrial practitioners, where VMs hosting performance tests are pre setup and the load drivers, with the same configurations, are setup in the VM. Practitioners will start the performance tests by initiating the load driver without really controlling the network traffic. We agree that leveraging a more controlled experiment is beneficial to further understand the discrepancy.

We clarify our experiments in the new draft and discuss the limitation of our approach. We also discuss the practice of conducting performance testing using virtual machines in order to better motivate the study and motivate our experimental choices. Please refer to comment R1.1 and R1.3. We put conducting a controlled experiment in our future plan as part of the conclusion of the paper.

Comment R3.43. *Lack of actionable findings. The paper repeats in several places that there is a discrepancy between the performance of virtual and physical environments. That is interesting, but it is not actionable. As a programmer or tester, I am not sure how to use this information. The paper does not give an explanation of how large the difference is, except by presenting the charts and figures. For example, in the last paragraph on page 9, "By looking closely at such metrics, we find..." Earlier it states that the "lines on the Q-Q plot are not close." I can see what the authors mean by looking at the figures, but is that a big difference? As a tester, I am already aware that the virtual environment is not a perfect reflection of a physical one. What I am wondering is, is the difference big enough that I need to worry about it? The paper does not quite answer that question. Perhaps the paper could go into an example to show how the results might affect decision-making, to give some context for understanding the results.*

Response. Thanks for pointing this out. We agree that between any two performance tests, not just between virtual and physical environments, there exist differences. This is exactly why we set the experiments in the contexts of 3 types of analyses of performance testing results. Our results measure the magnitude of the differences in a statistical manner, e.g., whether the difference is statistically significant, or how large is the relative differences. In addition, in our new revision, for every research question, we add a paragraph to discuss the impact of our results on practitioners interpretation of performance testing results using each type of analysis. Moreover, we also highlight the actionable implications at the end of each research question of in in our new revision.

Again, we thank all of you for your valuable feedback, which has made this a stronger manuscript. We look forward to hearing your feedback on the updated manuscript.

Sincerely,
Muhammad Moiz Arif, Weiyi Shang, & Emad Shihab

References

[1] BlackBerry. Blackberry enterprise server. `https://ca.blackberry.com/enterprise`, 2014. Accessed: 2017-04-04.

[2] Fabian Brosig, Fabian Gorsler, Nikolaus Huber, and Samuel Kounev. Evaluating approaches for performance prediction in virtualized environments. In *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2013 IEEE 21st International Symposium on*, pages 404–408. IEEE, 2013.

[3] Tse-Hsun Chen, Mark D Syer, Weiyi Shang, Zhen Ming Jiang, Ahmed E Hassan, Mohamed Nasser, and Parminder Flora. Analytics-driven load testing: An industrial experience report on load testing of large-scale systems.

[4] Ira Cohen, Moises Goldszmidt, Terence Kelly, Julie Symons, and Jeffrey S. Chase. Correlating instrumentation data to system states: A building block for automated diagnosis and control. In *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, OSDI'04, pages 16–16, 2004.

[5] Dee. performance-testing systems on virtual machines that normally run on physical machines. `http://sqa.stackexchange.com/questions/7709/performance-testing-systems-on-virtual-machines-that-normally-run-on-physical-ma`, 2014. Accessed: 2017-04-04.

[6] King Chun Foo, Zhen Ming Jiang, Bram Adams, Ahmed E Hassan, Ying Zou, and Parminder Flora. Mining performance regression testing repositories for automated performance analysis. In *Quality Software (QSIC), 2010 10th International Conference on*, pages 32–41. IEEE, 2010.

[7] Nikolaus Huber, Marcel von Quast, Michael Hauck, and Samuel Kounev. Evaluating and modeling virtualization performance overhead for cloud environments. In *CLOSER*, pages 563–573, 2011.

[8] Zhen Ming Jiang, Ahmed E Hassan, Gilbert Hamann, and Parminder Flora. Automated performance analysis of load tests. In *Software Maintenance, 2009. ICSM 2009. IEEE International Conference on*, pages 125–134. IEEE, 2009.

[9] Guoliang Jin, Linhai Song, Xiaoming Shi, Joel Scherpelz, and Shan Lu. Understanding and detecting real-world performance bugs. In *Proceedings of the 33rd ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '12, pages 77–88. ACM, 2012.

[10] Sean Kearon. Can you use a virtual machine to performance test an application? `http://stackoverflow.com/questions/8906954/can-you-use-a-virtual-machine-to-performance-test-an-application`, 2012. Accessed: 2017-04-04.

[11] Max Kuhn. Building predictive models in r using the caret package. *Journal of Statistical Software, Articles*, 28(5):1–26, 2008.

[12] H. Malik, Z. M. Jiang, B. Adams, A. E. Hassan, P. Flora, and G. Hamann. Automatic comparison of load tests to support the performance analysis of large enterprise systems. In *2010 14th European Conference on Software Maintenance and Reengineering*, pages 222–231, March 2010.

[13] Haroon Malik, Hadi Hemmati, and Ahmed E Hassan. Automatic detection of performance deviations in the load testing of large scale systems. In *Proceedings of the 2013 International Conference on Software Engineering*, pages 1012–1021. IEEE Press, 2013.

[14] Aravind Menon, Jose Renato Santos, Yoshio Turner, G John Janakiraman, and Willy Zwaenepoel. Diagnosing performance overheads in the xen virtual machine environment. In *Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments*, pages 13–23. ACM, 2005.

[15] Christopher L Merrill. Load testing sugarcrm in a virtual machine. `http://www.webperformance.com/library/reports/Virtualization2/`, 2009. Accessed: 2017-04-04.

[16] Thanh H.D. Nguyen, Bram Adams, Zhen Ming Jiang, Ahmed E. Hassan, Mohamed Nasser, and Parmin-

der Flora. Automated detection of performance regressions using statistical process control techniques. In *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering*, ICPE '12, pages 299–310, 2012.

[17] Thanh HD Nguyen, Bram Adams, Zhen Ming Jiang, Ahmed E Hassan, Mohamed Nasser, and Parminder Flora. Automated detection of performance regressions using statistical process control techniques. In *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering*, pages 299–310. ACM, 2012.

[18] A. Nistor, T. Jiang, and L. Tan. Discovering, reporting, and fixing performance bugs. In *2013 10th Working Conference on Mining Software Repositories (MSR)*, pages 237–246, May 2013.

[19] Adrian Nistor, Linhai Song, Darko Marinov, and Shan Lu. Toddler: Detecting performance problems via similar memory-access patterns. In *Proceedings of the 2013 International Conference on Software Engineering*, ICSE '13, pages 562–571, Piscataway, NJ, USA, 2013. IEEE Press.

[20] Weiyi Shang, Ahmed E. Hassan, Mohamed Nasser, and Parminder Flora. Automated detection of performance regressions using regression models on clustered performance counters. In *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering*, ICPE '15, pages 15–26. ACM, 2015.

[21] Eric Srion. The time for hyper-v pass-through disks has passed. `http://www.altaro.com/hyper-v/hyper-v-pass-through-disks/`, 2015. Accessed: 2017-04-04.

[22] SugarCRM. Sugarcrm. `https://www.sugarcrm.com/`, 2017. Accessed: 2017-04-04.

[23] Tintin. Performance test is not reliable on virtual machine? `https://social.technet.microsoft.com/Forums/windowsserver/en-US/06c0e09b-c5b4-4e2c-90e3-61b06483fe5b/performance-test-is-not-reliable-on-virtual-machine?forum=winserverhyperv`, 2011. Accessed: 2017-04-04.

[24] S. Tsakiltsidis, A. Miranskyy, and E. Mazzawi. On automatic detection of performance bugs. In *2016 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pages 132–139, Oct 2016.

[25] Pengcheng Xiong, Calton Pu, Xiaoyun Zhu, and Rean Griffith. vperfguard: an automated model-driven framework for application performance diagnosis in consolidated cloud environments. In *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering*, pages 271–282. ACM, 2013.

[26] S. Zaman, B. Adams, and A. E. Hassan. A qualitative study on performance bugs. In *2012 9th IEEE Working Conference on Mining Software Repositories (MSR)*, pages 199–208, June 2012.