Performance Testing

on Production System

Abstract

Performance testing is conducted to check whether the target application will be able to meet the real users' expectations in the production environment or not? Performance test results largely depend upon the test environment and having the test environment as an exact replica of the production system is the fundamental point in performance testing. Setting up such test environment is an extremely challenging task as the test results can be significantly affected if there are even any minor differences between the test and production environment. Quite often, performance testing is conducted on the production system to overcome the test environment issues. But again, testing on production system is also not straight forward and it has its own challenges. Most important challenge for the performance testing teams is to minimize the impact of production system performance testing on real users' activities and to completely test all the application bottlenecks as well.

In this paper, we will discuss what production system is, reasons to conduct performance testing on production system, myths regarding testing on production system, risks associated with testing on production system and best practices of effectively testing an application in the production environment.

Introduction and Background Information

It's always recommended to test the performance of Application Under Test (AUT) in a fully controlled lab environment which should be an exact replica of the production system. You can effectively test the application in such environment with simulated virtual users, automated framework and test data which can be refreshed at any time. In controlled environment, you test the target application under different load conditions unless the AUT is crashed. Although it's best to test the target application in a fully controlled lab environment but simultaneously it's very difficult to create the performance test environment that is an exact replica of the production system due to various challenges like replicating servers' infrastructure, network infrastructure, number of application tiers and database size etc.

An alternative option for all the challenges mentioned above is to test the application in production environment. Load testing on production system requires a different approach altogether with its own challenges and limitations. While testing in production environment, you need to design, execute and monitor the test very carefully to minimize its impacts on real users' activities. Simultaneously, you need to perform this exercise under sufficient amount of load to detect all AUT bottlenecks in order to efficiently complete the activity.

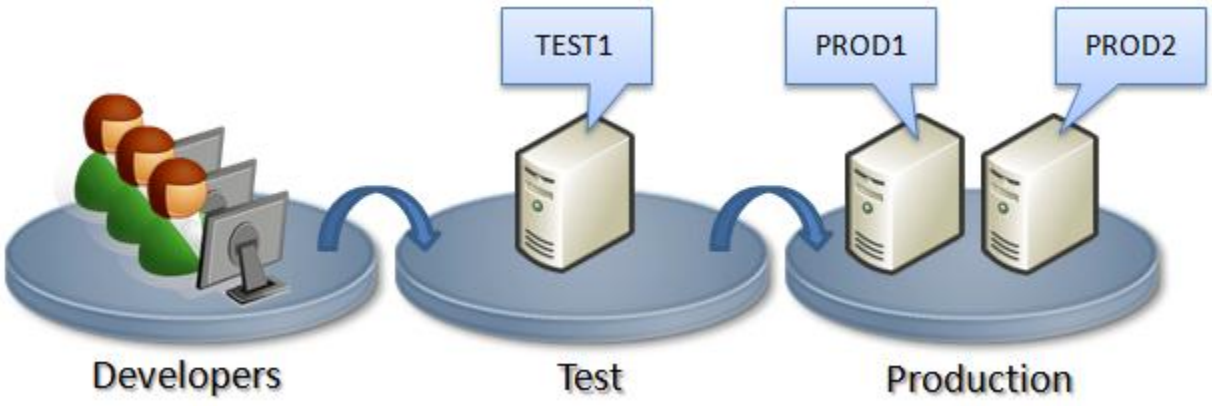
Furthermore, it can be extremely risky and erroneous to compare the lab environment test results to production system. You can never replicate 100% production system in lab environment and your test results are always affected due to the difference between the two environments. It will be a disaster to test the target application in such a lab environment that is different from the production system and then doing the production planning based on these results. It's the core responsibility of performance engineers to thoroughly review the production and test environments and list down all the differences and their possible impacts on the test results for a better results analysis. They should further communicate all these concerns to the stakeholders so that all the misconceptions and inconveniences can be avoided.

What is Production Environment?

It's very important to develop a clear understanding of the production/live environment before discussing the details regarding its performance testing.

Production environment is also called the application live environment. This is a set of resources which provides live services to real application users. In short, the environment on which live users interact is called the production/live environment.

Usually the following three different application environments are used in application lifecycle for its smooth and error-free working and limiting the impacts of development and testing teams and actual application users' on each other.



Development Environment: The set of resources accessed by the development team to build the application.

Test Environment: The set of resources on which the testing team works to test the application.

Production/Live Environment: The set of resources accessed by application users to perform their transactions.

Risks of Performance Testing on Production Environment

Although occasionally conducting performance testing on production system is the only way to successfully perform the activity, but still there are lots of risks associated with this approach. Testing on production system is the best method as well because it provides the application and its infrastructure in depth performance results. But you also need to analyze all its risks and have to make a decision based on your need of testing on production system.

Following is a brief list of risks associated with conducted performance testing on production environment,

- Actual users can experience major slowdown in application response
- Real users may not be able to complete their business transactions due to the slow response time
- Application can be slow even after test completion due to the data generated during the performance test execution
- Real user can start experiencing application errors and even the application can stop responding
- It will be difficult to identify the root-cause of the performance bottlenecks in the presence of real users along with simulated users load
- Real users need to stop the work on the application to get accurate test results but it will make the application unavailable during this time, which might not be possible on business critical applications

Reasons to Test on Production System

Most of the companies don't conduct their application performance testing due to the cost, resources and efforts involved in successfully performing the activity. But it can lead to a disaster once the application is in production along with the feeling of insecurity about its performance.

Moreover, it's hard to achieve the required test results if you are conducting performance testing in the test environment only. Various parts of the application and its infrastructure are not tested at all in the test environment and in order to test those missing areas you need to conduct testing on production system as well. Some of those missing areas are as follows:

- Third party Content Deliver Network (CDN) performance is not tested
- Firewall effects on application performance are not tested
- Application load balancing is not tested in test environment
- Application internet connection performance is not tested
- DNS lookup time is not tested in test lab

Although it has been discussed earlier that the testers face a lot of risks while conducting the performance testing on production system but still it has its own importance and is widely exercised especially when the production system is very complex and creating its exact replica is even more difficult. There are numerous other factors as well which motivate to conduct the performance testing on production system. Some of them are discussed below in detail.

Test Results Validation

Mostly performance testing is conducted on a clone of the target application which may not be 100% replica of the production system as mentioned above. Complete application and its infrastructure can't be tested in the test environment and the missing areas can significantly alter the test results. Other than these missing areas, the software, hardware and services used in the test environment can also differ from the production system. So based on these differences, the test environment results cannot be mapped directly on the production system. We need to validate the test environment results on the production system to get the real insights of the AUT and also to get a complete degree of confidence on the application performance in the production environment.

Cost Effective

Performance testing is indeed a costly and time taking activity especially when you need to replicate the production system in a test lab. Each application consists of various software (operating system, application servers, databases etc.), hardware (firewalls, routers, load balancers, servers etc.) and services (Content Delivery Network, ad servers, credit card verification systems etc.) and replicating all these is a big challenge which involve lots of efforts, time and money. In fact it doesn't matter how much efforts you put in, creating all of these as exactly the same as the production system is nearly impossible. Sometimes performance testing teams use the scaled environment by utilizing some parts of the production system in test lab and extrapolate the achieved results to map them on the production system. But it's always difficult to find out the performance bottlenecks beyond the scaled environment. Therefore by conducting performance testing on production system, lots of efforts, cost and time can be saved and more efforts can be put on the execution and analysis of the test in order to achieve better results.

Testing for Database Back-ups and Post-Crash Recovery

Performance testing concepts are relatively complex and lots of people are not even aware of all the outcomes which should be achieved from the performance testing activity. Testing the database back-ups and restoration of the production system after crashing are important application performance areas but usually many companies ignore these areas and consequently they don't engage the production environment for performance testing. The purpose of the performance testing is not only to check how the application behaves under different load conditions but to check its post-crash recovery process as well.

These important performance areas need to be carefully addressed in order to gain full degree of confidence about all the application performance areas in production environment. Critical business applications' post-crash recovery time must be known and should be communicated to all its users. While conducting performance testing, all such post-crash issues should be identified and fixed so that they don't cause any trouble in the production environment and disturb the users during the peak working hours.

Easier Test Environment Set-up

Setting up the performance test environment similar to the production environment is one of the most challenging tasks in the whole testing activity. Replicating the complete application infrastructure including all its servers and network infrastructure, all application tiers, database and all its dataset etc is indeed a tough thing to do. But you don't need to bother about replicating all these sources when you are testing on the production system. Testing with dataset similar to the production system is also extremely important for getting accurate performance results but it requires a lot of effort. However, all these efforts can be saved by conducting performance testing on the production system.

Myths regarding Testing on Production System

In the above section, we discussed the reasons why companies should test their applications in production environment. Despite of all these advantages of testing on production system, most companies are still reluctant to conduct performance testing on production systems mainly due to the following myths or misconceptions.

Production Testing is Live Testing

The most common misconception about testing on production system is that it will affect the real users' activities and application data. Although many companies conduct performance testing when actual users are also interacting with the application but still there are few approaches that can be followed to conduct the performance testing on production system without affecting the actual users and application data like,

- Testing during maintenance window
- Testing before releasing the application
- Perform read only transactions

Production Testing is Too Risky

Performance testing on production system although looks very risky but in fact it's not that much risky as many companies are afraid of it. You can mitigate most of its risks by simply planning your activities correctly like,

- Careful selection of test objective
- By following the approaches (i.e. testing during maintenance window or before release etc) discussed above
- You must have complete monitoring information during the test so that you can determine and react to the impacts of the test on application

Testing on Production is all about breaking the site

It's a huge misconception about performance testing that its purpose is just to test the application in such conditions where it will definitely crash. Although stress and failure testing techniques are part of the performance testing but still there are lots of other performance testing types as well and their purpose is to test the application behavior under normal user loads. Some of these testing types are as follows:

Load Testing: Testing the application response time under various normal load conditions

Baseline Testing: Baseline for load which the application can handle is established while meeting the success criteria

Spike Testing: Test the application under certain spike load conditions as well within the application capacity

Endurance Testing: Test the application stability under normal user load for longer duration (e.g. 24 hours)

Diagnostic Testing: Test is run to verify the troubleshooting of any issues or to observe the effect of infrastructure changes

Performance Testing Takes Too Long

Although an application's complete performance testing cycle can take fair amount of time, but you don't need to engage the production system from start to test completion. Production system is mainly engaged during the execution of the test which is only a fraction of the complete performance testing

activity. All other performance testing activities like planning, script creation, scenario building, analysis and reporting have almost no impact on actual users and application data.

Production Testing is Too Difficult

Many companies don't go for the testing on production system because they believe it's a tough thing to do and can negatively impact the application if not done properly. Although they are right in evaluating the impacts of conducting performance testing on production system in an inappropriate manner but they are wrong about evaluating the complexity of production system testing. In this technologically advanced era, we are fortunate enough to be equipped with all the required advanced performance testing tools which provide the complete application insights and are very easy to use as well. Another better technique could be to start with fairly simple test scenarios (e.g. reading or navigating on the application) which are easy to simulate but can reveal many issues before moving towards more complex and advance test scenarios.

Production Testing isn't required if we conduct thorough Testing in lab

We have already mentioned this, testing in lab environment is never enough unless you have replicated 100% production system which is again almost impossible. In lab environment you can only test the application but can't do much with its infrastructure testing. Numerous unexpected issues arise on application broader infrastructure itself. There are many areas in application infrastructure like CDN, Load balancers, Firewall capacity, DNS routing etc. which can only be tested on the production system.

No need for Production Testing unless Application faces bottlenecks

Due to all the above mentioned misconceptions, companies try to avoid the production system testing as long as they can. They only go for production system testing unless it starts facing major performance bottlenecks and issues. This reactive approach can cost their businesses a lot more as compared to conducting production system performance testing on initial stages. Moreover, bottlenecks root-cause identification and their fixing takes more time, effort and cost when you do this in production. So it's always better to be proactive and avoid unwanted situations which can impact your business.

Production Testing is Too Costly

Performance testing activity can be extremely costly with traditional testing approaches. You need to build a test lab which should be similar to the AUT and that can cost heavily. Moreover, you would be needing lots of load injectors for simulating the required user loads and all this can cost you a lot. However all these efforts and costs of preparing the test lab can be saved by way of conducting the performance testing on production system and cloud platforms can be used to simulate the high user loads from different geographical locations. Another misconception is that the performance testing tools are very expensive. Although it was the case a few years back but now with the incorporation of many new performance testing organizations and plenty of open source performance testing tools this

issue has been resolved. You can check out the AgileLoad pricing model (http://www.agileload.com/Product/products_list.aspx). Price is very nominal and quite affordable based on the services we provide to its users.

Best Practices of Testing on Production System

Although the various reasons due to which it's important to conduct the performance testing on production system have been discussed however there are still lots of issues and concerns (some of them have been discussed in the above section) on the basis of which the companies are hesitant to go for it. In this section, we will discuss some of the best practices that can be opted to minimize the impacts of performance testing on production system.

Testing During Maintenance Window

Almost all the large organizations' applications go for scheduled maintenance and during that period of time they restrict their users from interacting with the application. You can co-ordinate with responsible teams and plan out your performance testing activity during this scheduled downtime without affecting actual users' experience.

Test before Release

One of the best approaches could be to test the application just before making it available for actual users. You can include application performance testing part of your release management plan to make sure that performance tests are always executed before releasing the application.

Test during Off-Hours of Off-Days

Conduct the performance testing during off-hours of the off-days if you are not left with any of the above two options. Minimum number of application actual users is affected on conducting the performance testing at this schedule. It not only helps in minimizing the impact of testing on real users activities but also in identifying the bottlenecks root-causes. The best and most suitable time considered for such approach is midnight Saturday or Sunday.

Test Read-only Transactions

Many companies don't prefer to do any testing activity on their production system due to the fear that the test data might get mixed with the actual applications users' data. Especially in case of business critical applications, companies are not willing to take even the minor risks. That is why production database is almost never used in testing and even if it's used, it's used only for the read-only operations. These simple transactions don't affect the application data but can reveal important performance bottlenecks.

Increase Load Gradually

One approach that could be exercised to minimize the impact of performance testing on real users is to increase the simulated users gradually unless the real users' transactions are within the acceptable threshold. We have mentioned above that performance testing is not all about breaking the system but also to find out the application behavior under normal conditions. Run a test and increase the load gradually unless the users' response time is within the acceptable range and they are able to successfully perform their transactions. Then analyze the test results, fix the bottlenecks and re-test. You can thoroughly test any application for most of its bottlenecks in multiple iterations without actually impacting the real users, however they will be experiencing slower user experience but still will be able to complete their transactions.

Careful Monitoring and Continuous Communication during Test Execution

The performance testing approach and its expected outcomes along with the involved risks should be clearly communicated to all the stakeholders. Moreover, you need to be very pro-active while testing on the production system and all the stakeholders should be carefully monitoring the test and test should be stopped immediately if and when it affects the actual users beyond their acceptable threshold.

Conclusion

Setting up the test environment completely similar to the production system is always essential to attain accurate AUT performance results. Due to the various application and infrastructure mapping challenges, it's almost impossible to setup the exact replica of the production system in a test lab. It's also recommended to test the production system as it provides the target application's insight performance results with less effort and cost along with the test environment results verification. Most companies avoid testing in production environment due to its impacts on actual users' activities and their data. There are numerous misconceptions regarding testing on production system like it's too risky, it will break the system, it takes too long, it's too costly and it's not required once we have done testing in lab etc. The impact of production system testing on actual application users can be minimized by following few safety measures like testing during off-hours of off-days, testing before release, testing during maintenance window, performing read only database operations and careful monitoring of the test execution. So the bottom line is that you must test your application on production system by following the above mentioned best practices to minimize its impacts on application and its actual users.